

# ВЕСТНИК

НАЦИОНАЛЬНОГО ТЕХНИЧЕСКОГО  
УНИВЕРСИТЕТА «ХПИ»

*Сборник научных трудов*

*Тематический выпуск «Системный анализ, управление и информационные технологии»* 9'2010

Издание основано Национальным техническим университетом «ХПИ»  
в 2001 году

Госиздание

Свидетельство Госкомитета по информационной политике Украины  
КВ № 5256 от 2.07.2001 г.

## КООРДИНАЦИОННЫЙ СОВЕТ

### Председатель

Л.Л. Товажнянский, д-р техн. наук, проф.

### Зам. председателя

А.П. Марченко, д-р техн. наук, проф.

Е.И. Сокол, д-р техн. наук, проф.

### Ответственный секретарь

К.А. Горбунов, канд. техн. наук, доц.

Е.Е. Александров, д-р техн. наук, проф.

Л.Л. Бесов, д-р ист. наук, проф.

Б.Т. Бойко, д-р техн. наук, проф.

Ф.Ф. Гладкий, д-р техн. наук, проф.

М.Д. Годлевский, д-р техн. наук, проф.

А.И. Грабченко, д-р техн. наук, проф.

В.Г. Данько, д-р техн. наук, проф.

В.Д. Дмитриенко, д-р техн. наук, проф.

И.Ф. Домнин, д-р техн. наук, проф.

В.В. Епифанов, канд. техн. наук, проф.

Ю.И. Зайцев, канд. техн. наук, проф.

П.А. Качанов, д-р техн. наук, проф.

В.Б. Клепиков, д-р техн. наук, проф.

С.И. Кондрашов, д-р техн. наук, проф.

В.М. Кошельник, д-р техн. наук, проф.

В.И. Кравченко, д-р техн. наук, проф.

Г.В. Лисачук, д-р техн. наук, проф.

В.С. Луников, д-р техн. наук, проф.

А.К. Морачковский, д-р техн. наук, проф.

В.И. Николаенко, канд. ист. наук, проф.

П.Г. Перерва, д-р экон. наук, проф.

В.Б. Самородов, д-р техн. наук, проф.

Г.М. Сучков, д-р техн. наук, проф.

Н.А. Ткачук, д-р техн. наук, проф.

В.А. Пуляев, д-р техн. наук, проф.

М.И. Рыщенко, д-р техн. наук, проф.

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

### Ответственный редактор

М.Д. Годлевский, д-р техн. наук, проф.

### Зам. ответственного редактора

А.С. Куценко, д-р техн. наук, проф.

### Ответственный секретарь

Н.И. Безменов, канд. техн. наук, проф.

И.П. Гамаюн, д-р техн. наук, проф.

В.Д. Дмитриенко, д-р техн. наук, проф.

А.Е. Ефимов, д-р техн. наук, проф.

И.В. Кононенко, д-р техн. наук, проф.

Л.М. Любчик, д-р техн. наук, проф.

В.П. Северин, д-р техн. наук, проф.

Л.Г. Раскин, д-р техн. наук, проф.

Н.В. Шаронова, д-р техн. наук, проф.

М.А. Ястребенецкий, д-р техн. наук, проф.

## АДРЕС РЕДКОЛЛЕГИИ

61002, Харьков, ул. Фрунзе, 21

Кафедры «Автоматизированные  
системы управления» и «Системный  
анализ и управление»

Тел. (057) 707-65-20,

(057) 707-61-03

УДК 681.5+519.7+519.8+004

**Вестник НТУ «ХПИ».** Сборник научных трудов. Тематический выпуск  
«Системный анализ, управление и информационные технологии». –  
Харьков: НТУ «ХПИ». – 2010. – № 9. – 204 с.

У збірнику представлено теоретичні та практичні результати наукових досліджень та розробок, що виконані викладачами вищої школи, аспірантами, студентами, науковими співробітниками, спеціалістами різних організацій та підприємств.

Для науковців, викладачів, аспірантів, студентів.

В сборнике представлены теоретические и практические результаты научных исследований и разработок, которые выполнены преподавателями высшей школы, аспирантами, студентами, научными сотрудниками, специалистами различных организаций и предприятий.

Для научных работников, преподавателей, аспирантов, студентов.

**Друкуються за рішенням Вченої ради НТУ «ХПИ»,  
протокол № 13 від 25.12.2009 р.**

*С. М. БАЛЮТА*, канд. техн. наук, доцент НУПТ,  
*О. Н. БЕЗМЕНОВА*, студентка НТУ «ХПИ»

### КВАЗИСТАТИЧЕСКАЯ МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЗАДАЧИ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПАРАМЕТРАМИ ПРОКАТНОГО СТАНА

У статті запропоновано квазістатичну математичну модель металопрокатного стану, що дозволяє звести задачу вибору настроювальних технологічних параметрів до класичної двоточковою крайовою задачею для системи звичайних диференціальних рівнянь. Математична модель може бути використана також для синтезу системи стабілізації обраних технологічних параметрів.

В статье предложена квазистатическая математическая модель металопрокатного стана, позволяющая свести задачу выбора настроечных технологических параметров к классической двухточечной краевой задаче для системы обыкновенных дифференциальных уравнений. Математическая модель может быть использована также для синтеза системы стабилизации выбранных технологических параметров.

In the article the quasi-static mathematical model of the metal-rolling mill is proposed, which makes it possible to reduce the tuning technological parameters selection problem to the classical two-point boundary value problem for the system of ordinary differential equations. The mathematical model can be also used for the synthesis of stabilizing system of selected technological parameters.

**Введение.** Современный металлопрокатный стан представляет собой один из сложнейших объектов управления. Эта сложность обусловлена множеством различных по природе взаимосвязанных между собой подпроцессов, составляющих процесс прокатки. К ним относятся, прежде всего, процессы термомеханической деформации металлов, механические процессы в элементах клеток, а также электромеханические процессы в системах приводов. Кроме перечисленных подпроцессов, можно отметить и другие, детально рассмотренные в соответствующей научно-технической литературе [1, 2].

Математические модели отдельных подпроцессов металлопрокатного производства (МПП) могут быть линейными, нелинейными, распределенными, дискретными, описываться алгебраическими или дифференциальными уравнениями. Такое разнообразие классов математических моделей приводит к серьезным затруднениям при синтезе систем автоматического управления МПП. Кроме того, математические модели некоторых элементарных процессов МПП являются весьма приближенными по структуре и содержат ряд неопределенных параметров.

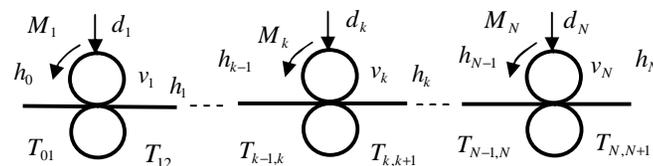
Указанные обстоятельства приводят к необходимости рассматривать упрощенные математические модели МПП, позволяющие синтезировать приближенные законы управления. Уточнение процесса управления осуществляется на основе стабилизирующих воздействий по отклонению от

программной траектории. Такая двухуровневая концепция построения систем управления рассмотрена в [3].

Особенный интерес с практической точки зрения представляет математическая модель МПП, основанная на квазистатическом подходе, известном из классической термодинамики. При таком подходе реальные фазовые траектории МПП, полученные в результате изменения управляющих параметров, заменяются последовательностями положений равновесия, соответствующих последовательностям значений управляющих воздействий. При определенных условиях на скорость изменения последних такое представление управляемого процесса дает достаточно хорошее приближение к реальному. Такие условия для класса линейных систем рассмотрены в [4]. Основным достоинством квазистатического моделирования управляемых процессов является возможность построения математической модели на основании статических взаимосвязей «вход–выход», которые могут быть получены в результате экспериментальных исследований.

Целью настоящей работы является разработка квазистатической математической модели МПП и анализ на ее основе различных путей решения задачи перенастройки технологических параметров.

**Математическая модель МПП.** Рассмотрим упрощенную структурную схему процесса холодной прокатки металла в многоклетевом непрерывном стане, изображенную на рисунке.



Структурная схема процесса прокатки

На рисунке введены следующие обозначения основных технологических параметров МПП:

$h_0, h_1, \dots, h_N$  – толщины полосы,

$v_1, v_2, \dots, v_N$  – скорости полосы,

$T_{01}, T_{12}, T_{23}, \dots, T_{N,N+1}$  – натяжения полосы,

$M_1, M_2, \dots, M_N$  – моменты прокатки на различных участках прокатного стана,

$d_1, d_2, \dots, d_N$  – перемещения нажимных винтов.

Толщина полосы и момент прокатки на  $k$ -м участке связаны с остальными технологическими параметрами некоторыми соотношениями [4]:



общем случае будем считать, что допустимая в соответствии с указанными условиями область представляет собой выпуклый многогранник  $L$ .

Таким образом, задачу начальной настройки технологического процесса можно сформулировать в алгебраическом виде: найти вектор  $x \in M \cap L$ , или, проще говоря, найти какое-либо решение системы алгебраических уравнений (6), удовлетворяющее условию  $x \in L$ .

Поскольку таких векторов  $x$  в общем случае может быть множество  $X^*$ , то естественно сформулировать задачу оптимизации начальной настройки технологического процесса в следующем виде: найти вектор технологических параметров  $x \in X^*$ , доставляющий минимум целевой функции

$$z = F(x_1, x_2, \dots, x_n). \quad (7)$$

Задачу оптимизации можно свести к последовательности решений задач начальной настройки, включив (7) при некотором фиксированном значении  $z$  в систему уравнений (6) и уменьшая на каждом шаге величину  $z$  в (7). Такой подход отличается от традиционного, основанного на оптимизации функций (7) при условиях (6) и  $x \in L$  методами множителей Лагранжа или штрафных функций, и по своей сути сводится к одномерному поиску экстремума на решениях расширенной системы алгебраических уравнений (6), (7).

**Квазистатический метод решения задачи начальной настройки.** Сформулированная выше задача начальной настройки параметров технологического процесса представляет собой задачу нахождения одного из решений системы (6), удовлетворяющего условию  $x \in L$ . Одним из возможных путей ее решения является сведение к задаче минимизации целевой функции

$$F(x) = \sum_{k=1}^m f_k^2(x), \quad x \in L. \quad (8)$$

Однако, численное решение задачи минимизации (8), как правило, затруднительно в связи с наличием большого количества локальных минимумов и «оврагов» [7].

Ниже будет предложен принципиально иной подход к решению сформулированной задачи, основанной на решении эквивалентной задачи управления квазистатическим процессом, соответствующим системе (6).

Введем в рассмотрение вектор невязок  $y \in R^m$ :

$$\begin{aligned} y_1 &= f_1(x_1, \dots, x_n), \\ y_2 &= f_2(x_1, \dots, x_n), \\ &\dots\dots\dots \\ y_m &= f_m(x_1, \dots, x_n), \end{aligned} \quad (9)$$

или в векторной форме

$$y = f(x^1, x^2) \quad (10)$$

и будем рассматривать векторы  $x$  и  $y$  как некоторые функции времени.

Продифференцируем (10) по времени. В результате получим

$$\frac{\partial f}{\partial x^1} \dot{x} + \frac{\partial f}{\partial x^2} \dot{x} = \dot{y} \quad (11)$$

Обозначая  $\frac{\partial f}{\partial x^1} = W(x)$ ,  $\frac{\partial f}{\partial x^2} = V(x)$ , где  $W(x)$  и  $V(x)$  – матрицы Якоби размерностей  $n \times m$  и  $m \times r$ ,  $\dot{x} = v$ ,  $\dot{y} = u$ , представим математическую модель настройки технологических параметров в виде управляемой системы дифференциальных уравнений

$$\begin{aligned} \dot{x} &= \Phi(x)u - \Phi(x)V(x)v, \\ \dot{x} &= v, \\ \dot{y} &= u, \end{aligned} \quad (12)$$

в которой  $(x^T, y^T)^T$  – вектор состояний, а  $v$  и  $u$  – векторы управлений,  $\Phi(x) = W^{-1}(x)$ .

Таким образом, задачу настройки технологического процесса можно свести к следующей задаче управления: из некоторого начального состояния  $(x_0^T, y_0^T)^T$  перевести систему (12) в конечное состояние  $(x_L^T, 0)^T$ , где  $x_L \in L$ ,  $0$  –  $(1 \times r)$ -вектор. Начальное состояние выбирается путем вычисления вектора  $y_0$  в результате подстановки некоторого начального значения  $x_0$  в (9) т.е.  $y_0 = f(x_0)$ . Будем также предполагать, что множество  $L$  представляет собой сумму множеств  $L = L_1 + L_2$ , причем  $x^1 \in L_1$ ,  $x^2 \in L_2$ . Отметим также, что, поскольку временной масштаб не задан, то можно принять временной интервал единичным  $t \in [0, 1]$ , а значения  $u$  и  $v$  неограниченными.

Рассмотрим задачу управления квазистатической системой (12) при фиксированном  $x^2 \in L_2$ . В этом случае, задача управления примет упрощенный вид: для управляемой системы

$$\begin{aligned} \dot{x} &= \Phi(x^1, x^2)u, \\ \dot{y} &= u, \end{aligned} \quad (13)$$

найти закон управления  $\mathbf{u}(t)$ ,  $t \in [0,1]$ , при котором  $\mathbf{y}(1) = \mathbf{0}$ , где  $\mathbf{0}$  –  $(n \times 1)$ -вектор. Нетрудно видеть, что одним из таких законов будет

$$\mathbf{u}(t) = -\mathbf{y}_0 = \mathbf{c}. \quad (14)$$

Тогда искомое значение  $\mathbf{x}^1(1)$  может быть получено в результате интегрирования системы дифференциальных уравнений

$$\dot{\mathbf{x}} = \Phi(\mathbf{x}^1, \mathbf{x}^2) \mathbf{c}. \quad (15)$$

при фиксированном векторе  $\mathbf{x}^2$  и начальном значении  $\mathbf{x}^1(0)$ , полученном ранее.

Интегрирование (15) не вызывает принципиальных затруднений, если матрица  $\Phi(\mathbf{x})$  получена аналитически путем обращения матрицы Якоби  $\mathbf{W}(\mathbf{x})$  в общем виде. Обойти операцию обращения матрицы  $\mathbf{W}(\mathbf{x})$  предлагается следующим образом. Воспользуемся известной формулой для производной обратной матрицы

$$\dot{\Phi} = -\Phi \mathbf{W} \Phi. \quad (16)$$

В свою очередь полная производная матрицы-функции  $\mathbf{W}(\mathbf{x}^1, \mathbf{x}^2)$  по времени может быть представлена в виде

$$\dot{\mathbf{W}}(\mathbf{x}^1, \mathbf{x}^2) = \sum_{k=1}^m \frac{\partial \mathbf{W}}{\partial x_k} (\boldsymbol{\varphi}_k, \mathbf{c}), \quad (17)$$

где  $\boldsymbol{\varphi}_k$  –  $k$ -й столбец матрицы  $\Phi$ .

Тогда процедуру обращения матрицы  $\mathbf{W}(\mathbf{x})$  на каждом шаге интегрирования системы (15) можно заменить шагом интегрирования матричного дифференциального уравнения (16), обратив матрицу  $\mathbf{W}(\mathbf{x})$  только один раз при значении  $\mathbf{x} = \mathbf{x}_0$ .

**Сведение к двухточечной краевой задаче.** Рассмотренный выше подход к нахождению настройки технологических параметров позволяет организовать управление настройкой путем изменения вектора параметров  $\mathbf{x}^2 \in L_2$ , так чтобы решение дифференциального уравнения (15) удовлетворяло условию  $\mathbf{x}^2(1) \in L_1$ . Для организации целенаправленного изменения вектора  $\mathbf{x}^2$  вернемся к исходной системе (12). Ее структура такова, что только выбором вектора управлений  $\mathbf{u}(t)$  можно получить нулевую невязку  $\mathbf{y}(1)$ . Например, в виде (14). Вектор управлений  $\mathbf{v}(t)$  обеспечивает в свою очередь конечные

условия на управляемый процесс  $\mathbf{x}^1(1) \in L_1$ ,  $\mathbf{x}^2(1) \in L_2$  при условии выполнения  $\mathbf{y}(1) = \mathbf{0}$ .

Для того чтобы использовать классические методы теории управления введем удобный для дальнейшего рассмотрения критерий оптимальности

$$J = \frac{1}{2} \int_0^1 \mathbf{v}^T \mathbf{v} dt, \quad (18)$$

а вместо многогранников  $L_1$  и  $L_2$  будем рассматривать их приближения в форме эллипсоидов  $E_1$  и  $E_2$  [5]. Такая замена цели управления позволяет воспользоваться традиционной формой условий трансверсальности вариационного исчисления и принципа максимума Понтрягина. С учетом сделанных предположений функция Гамильтона примет вид

$$H = \frac{1}{2} \mathbf{v}^T \mathbf{v} + \boldsymbol{\lambda}_1^T \Phi \mathbf{c} - \boldsymbol{\lambda}_1^T \Phi \mathbf{V} \mathbf{v} + \boldsymbol{\lambda}_2^T \mathbf{v}, \quad (19)$$

где  $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1^T, \boldsymbol{\lambda}_2^T)^T$  – вектор сопряженных переменных.

В соответствии с принципом максимума оптимальный закон управления найдем из условия

$$\frac{\partial H}{\partial \mathbf{v}} = \mathbf{v}^T - \boldsymbol{\lambda}_1^T \Phi \mathbf{V} + \boldsymbol{\lambda}_2^T = \mathbf{0},$$

где  $\mathbf{0}$  –  $(r \times 1)$ -вектор.

Таким образом, оптимальный закон управления примет вид

$$\mathbf{v}^T = \boldsymbol{\lambda}_1^T \Phi \mathbf{V} - \boldsymbol{\lambda}_2^T. \quad (20)$$

В свою очередь вектор сопряженных переменных удовлетворяет системе дифференциальных уравнений

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} = -\frac{\partial}{\partial \mathbf{x}} (\boldsymbol{\lambda}_1^T \Phi(\mathbf{x}) \mathbf{c} - \boldsymbol{\lambda}_1^T \Phi(\mathbf{x}) \mathbf{V}(\mathbf{x}) \mathbf{v}) \quad (21)$$

образующей совместно с первыми двумя уравнениями (12) гамильтонову систему  $2n$  дифференциальных уравнений.

Сопряженную систему (21) можно представить в следующем виде:

$$\dot{\boldsymbol{\lambda}} = -\boldsymbol{\lambda}_1^T \frac{\partial \Phi}{\partial x_k} \mathbf{c} + \boldsymbol{\lambda}_1^T \frac{\partial \Phi}{\partial x_k} \mathbf{V} \mathbf{v} + \boldsymbol{\lambda}_1^T \frac{\partial \mathbf{V}}{\partial x_k} \mathbf{v}, \quad k = \overline{1, n}. \quad (22)$$

В свою очередь матрицы  $\frac{\partial \Phi}{\partial x_k}$  могут быть выражены через производную от матрицы  $W$  по аналогии с (16):

$$\frac{\partial \Phi}{\partial x_k} = -\Phi \frac{\partial W}{\partial x_k} \Phi. \quad (23)$$

Постановка двухточечной краевой задачи завершается формулировкой условий трансверсальности состоящих в ортогональности сопряженных векторов  $\lambda_1$  и  $\lambda_2$  эллипсоидам  $E_1$  и  $E_2$  соответственно в момент времени  $t=1$

$$\lambda_1^T = c_1 \frac{\partial E_1}{\partial x^1}, \quad \lambda_2^T = c_2 \frac{\partial E_2}{\partial x^2}, \quad (24)$$

где  $c_1$  и  $c_2$  – неопределенные постоянные.

Таким образом, решение двухточечной краевой задачи состоит в нахождении начальных условий для сопряженной системы (22) таких, чтобы решения гамильтоновой системы (12) и (22) с учетом закона управления (20) в конечный момент времени  $t=1$  удовлетворяли условиям  $x \in L$  и условиям (24).

Решение сформулированной двухточечной краевой задачи может быть получено в результате применения численных методов, изложенных в соответствующей литературе [6].

**Заключение.** Основным результатом исследования является разработка квазистатической математической модели металлопрокатного стана. Достоинством предложенной математической модели является ее относительная простота, основанная на статических соотношениях между технологическими параметрами. С другой стороны система дифференциальных уравнений, описывающая квазистатические фазовые траектории, позволяет применить мощный арсенал средств теории оптимального управления для решения различных задач, связанных с перенастройкой технологических параметров процесса прокатки.

**Список литературы:** 1. Целиков А.И. Теория расчета усилий в прокатных станах. – М.: Металлургиздат, 1962. – 434с. 2. Дружинин Н.Н. Непрерывные станы как объект автоматизации. – М.: Металлургия, 1967. – 259с. 3. Моисеев Н.Н. Математические задачи системного анализа. – М.: Наука, 1981. – 488с. 4. Куценко А.С., Чан Занг Лю Критерии адекватности динамических и статических математических моделей технологических процессов// Вісник Національного технічного університету «ХПІ». – Харків: НТУ «ХПІ». – 2003. – № 18. – С.23–28. 5. Черноусько Ф.Л. Оценка фазового состояния динамических систем. Метод эллипсоидов. – М.: Наука, Главная редакция физико-математической литературы, 1988. – 320с. 6. Брайсон А., Хо-Ю-Ши Прикладная теория оптимального управления. – М.: Мир, 1972. – 544с. 7. Калиткин Н.Н. Численные методы. – М.: Наука, Главная редакция физико-математической литературы, 1978. – 512с.

Поступила в редакцию 18.12.09

УДК 519.677:658.23

*Н.В. ШАТОХИНА*, канд. техн. наук, доцент НТУ «ХПИ»

## МОДЕЛИРОВАНИЕ ЭФФЕКТИВНЫХ ПАРАМЕТРОВ МОДЕЛИ ДЕЯТЕЛЬНОСТИ ПРОИЗВОДСТВЕННО-ЭКОНОМИЧЕСКОЙ СИСТЕМЫ

В статті запропоновано математичний апарат оптимізації показників діяльності виробничо-економічної системи як розв'язання задачі синтезу власних векторів відповідних матриць.

В статье предложен математический аппарат оптимизации показателей деятельности производственно-экономической системы как решение задачи синтеза собственных векторов соответствующих матриц.

In this article the mathematical apparatus for optimization of production and business system parameters as solution for eigenvectors synthesis task of correspondent matrixes have been proposed.

**Введение.** В настоящее время математические методы и модели находят широкое применение в стратегическом планировании и управлении. Полезные прикладные результаты получены при моделировании производственных, экономических и организационных процессов [1].

Производственное предприятие, организация или некоторый проект представляют собой сложную систему. Функционирование такой системы происходит под воздействием целого набора различных факторов, постоянно изменяющихся под воздействием внешних условий. Очевидно, что управление подобной системой представляет нетривиальную задачу.

Решение подобных задач становится практически невозможным без внедрения информационных систем, основной составляющей которых являются информационные технологии, основанные на применении совокупности моделей и методов, разработанных при помощи некоторого математического аппарата, и средств обработки информации.

Разработка и анализ результатов использования экономико-математических моделей позволяет не только ускорить процесс принятия решений, но и более комплексно представить рассматриваемую проблему по сравнению с анализом, который может провести даже самый высококвалифицированный и опытный эксперт.

Эффективность экономико-математического моделирования в значительной степени определяется адекватностью построенной модели, а также правильным выбором методологий и методов работы с ней.

Анализ используемых до настоящего времени на практике подходов к моделированию факторов, влияющих на функционирование организации, показывает, что большинство из них носят скорее качественный характер [2]. Это затрудняет их формализацию и автоматизацию учета.

В данной статье предложен математический аппарат, позволяющий моделировать воздействие внешних факторов на внутренние показатели деятельности организации и определять оптимальные значения рассматриваемых параметров модели для обеспечения эффективного функционирования системы, её адаптации к изменениям во внешней среде и развитию.

**1. Оптимизация внутренних экономических показателей как задача синтеза собственных векторов соответствующих матриц.** Предположим, что по результатам проведенного анализа рынка, прогнозам сформирован некоторый набор факторов, воздействующих на рассматриваемую систему. Обозначим эти параметры как элементы матриц  $\mathbf{A}$ ,  $\mathbf{B}$  соответственно  $a_{ji}$ ,  $b_{ji}$  ( $i, j = \overline{1, n}$ ). Пусть деятельность предприятия характеризуется набором показателей  $x_{ji}$ , которые будем трактовать как компоненты  $n$  векторов  $\mathbf{x}_i$ . Рассмотрим вначале случай, когда установлены оптимальные значения этих показателей для  $i$ -го вектора  $-\mathbf{x}_i^*$ . Задача формулируется так: необходимо определить значения части параметров  $a_{ji}$ ,  $b_{ji}$ , в дальнейшем их удобно обозначить как  $p_k$  ( $k = \overline{1, m}$ ), при которых будут достигаться оптимальные значения показателей деятельности.

Полагая зависимости между параметрами линейными, модель представим в форме задачи о собственных значениях и собственных векторах

$$(-\mathbf{A}\lambda + \mathbf{B})\mathbf{x} = \mathbf{0}. \quad (1)$$

Для решения оптимизационной задачи воспользуемся идеей теории чувствительности [3]. Варьируя параметры модели, придадим соответствующему собственному вектору необходимую конфигурацию. Обозначим изменение  $\mathbf{x}_i$  через

$$\Delta \mathbf{x}_i = \mathbf{x}_i^* - \mathbf{x}_i \quad (i = \overline{1, n}). \quad (2)$$

Введем в рассмотрение частные производные  $\frac{\partial \mathbf{x}_i}{\partial p_k}$  вектора  $\mathbf{x}_i$  по варьируемым параметрам  $p_k$  ( $k = \overline{1, m}$ ). Теперь вектор  $\Delta \mathbf{x}_i$ , полагая его достаточно малым, с точностью до малых второго порядка можно представить в виде.

$$\Delta \mathbf{x}_i = \sum_{k=1}^m \frac{\partial \mathbf{x}_i}{\partial p_k} \Delta p_k. \quad (3)$$

где  $\Delta p_k$  – изменение  $k$ -го параметра.

Для  $\frac{\partial \mathbf{x}_i}{\partial p_k}$  справедливо разложение

$$\frac{\partial \mathbf{x}_i}{\partial p_k} = \sum_{j=1}^n d_{ij} \mathbf{x}_j. \quad (4)$$

Если выражение (1) про дифференцировать по параметру  $p_k$  ( $k = \overline{1, m}$ ) и затем умножить слева на  $\mathbf{x}_l^T$  ( $l \neq i$ ), то с учетом формулы (4) получим

$$d_{ilk} = \frac{\mathbf{x}_l^T \left( \frac{\partial \mathbf{B}_k}{\partial p_k} - \lambda_i \frac{\partial \mathbf{A}_k}{\partial p_k} \right) \mathbf{x}_i}{(\lambda_i - \lambda_l) \mathbf{x}_l^T \mathbf{A} \mathbf{x}_l}. \quad (5)$$

Для  $l=i$  полагаем  $d_{ilk} = 0$ .

С учетом соотношения (4) выражение (3) приобретает вид

$$\Delta \mathbf{x}_i = \mathbf{S} \Delta \mathbf{p}. \quad (6)$$

где  $\mathbf{S}$  – матрица чувствительности,

$$\mathbf{S} = \begin{bmatrix} \sum_{j=1}^n d_{ij1} x_{1j} & \mathbf{L} & \sum_{j=1}^n d_{ijm} x_{1j} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \sum_{j=1}^n d_{ij1} x_{nj} & \mathbf{L} & \sum_{j=1}^n d_{ijm} x_{nj} \end{bmatrix}; \quad (7)$$

$\Delta \mathbf{p} = [\Delta p_1, \mathbf{K}, \Delta p_m]^T$  – вектор изменения параметров.

Формулу (6) можно рассматривать как уравнение относительно  $\Delta \mathbf{p}$ . В зависимости от соотношения  $n$  и  $m$  для  $\Delta \mathbf{p}$  имеют место следующие выражения:

$$\Delta \mathbf{p} = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \Delta \mathbf{x}_i \quad (m < n); \quad (8)$$

$$\Delta \mathbf{p} = \mathbf{S}^{-1} \Delta \mathbf{x}_i \quad (m = n); \quad (9)$$

$$\Delta \mathbf{p} = \mathbf{S}^T (\mathbf{S} \mathbf{S}^T)^{-1} \Delta \mathbf{x}_i \quad (m > n). \quad (10)$$

Формула (8) получена методом наименьших квадратов. С помощью выражения (10) из бесчисленного множества решений выделяется единственное, обладающее свойством  $\|\Delta \mathbf{p}\|_2 \Rightarrow \min$ .

Новое значение вектора параметров, отвечающего измененному собственному вектору, приближенно представим в виде

$$\mathbf{p} = \mathbf{p}_0 + \Delta \mathbf{p}, \quad (11)$$

где  $\mathbf{p}_0$  – исходный вектор параметров.

В формуле (3)  $\Delta \mathbf{x}_i$  предполагается малым, что практически редко выполняется. Для больших значений  $\Delta \mathbf{x}_i$  требуется разработка специального алгоритма, позволяющего определить необходимые изменения параметров с помощью одной из формул (8)-(10). Если модули отдельных координат в векторе  $\Delta \mathbf{x}_i$  велики, то первый шаг выполняется с  $\Delta \tilde{\mathbf{x}}_i = \frac{\Delta \mathbf{x}_i}{r}$  ( $r=1,2,\mathbf{K}$ );  $r$

выбирается так, чтобы модуль  $\Delta x_{ji}$  ( $j=1, n$ ) был небольшим, например, менее 5% от модуля  $x_{ji}$ . На каждом шаге реализуется итерационный процесс.

После первой итерации по формуле (11) получаем вектор параметров  $\mathbf{p}^{(1)}$  и новый собственный вектор  $\mathbf{x}_i^{(1)}$ . В общем случае последний отличается от вектора  $\mathbf{x}_i + \Delta \mathbf{x}_i$ , т.е. вектора, который мы стремимся получить на первом шаге. Определяем невязку  $\Delta \tilde{\mathbf{x}}_i^{(1)} = \mathbf{x}_i + \Delta \tilde{\mathbf{x}}_i - \mathbf{x}_i^{(1)}$  и выполняем с ней вторую итерацию для нахождения  $\mathbf{p}^{(2)}$ ,  $\mathbf{x}_i^{(2)}$ . Затем вычисляем невязку  $\Delta \tilde{\mathbf{x}}_i^{(2)} = \mathbf{x}_i + \Delta \tilde{\mathbf{x}}_i - \mathbf{x}_i^{(2)}$ . Процесс продолжается до тех пор, пока не выполнится условие

$$\frac{|p_k^{(j+1)} - p_k^{(j)}|}{|p_k^{(j)}|} < \varepsilon \quad (k=1, m),$$

где  $\varepsilon$  – некоторое малое число.

Полученный на последней итерации собственный вектор  $\mathbf{x}_i^{(j+1)}$  принимаем за вектор  $\mathbf{x}_i$ , с его помощью уточняем  $\Delta \mathbf{x}_i$  по формуле (2) и выполняем второй шаг с  $\Delta \tilde{\mathbf{x}}_i = \frac{\Delta \mathbf{x}_i}{r-1}$  аналогично первому. Последний  $r$ -й шаг, очевидно, выполняется для

$$\Delta \tilde{\mathbf{x}}_i = \frac{\Delta \mathbf{x}_i}{r - (r-1)} = \Delta \mathbf{x}_i.$$

Отметим несколько обстоятельств, важных при программной реализации алгоритма. В программе необходимо предусмотреть автоматическое изменение  $r$ , поскольку в некоторых случаях даже малое изменение собственного вектора требует значительного изменения некоторых параметров. Если знак добавки отрицательный, новое значение параметра также может быть отрицательным, т.е. лишенным физического смысла. Исходный улучшенный собственный вектор  $\mathbf{x}_i^*$ , а также улучшенные собственные векторы на каждом шаге алгоритма следует предварительно нормировать, причем нормировка должна использоваться та же, что и в подпрограмме определения собственных значений и собственных векторов.

Изложенный алгоритм без труда обобщается на случай одновременной корректировки всех или нескольких собственных векторов. Полученная выше формула (6) приобретает вид

$$\Delta \mathbf{x} = \mathbf{S} \Delta \mathbf{p}. \quad (12)$$

Причем вектор  $\Delta \mathbf{x}$  и матрицу  $\mathbf{S}$  следует теперь рассматривать как блочную

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta \mathbf{x}_1 \\ \Delta \mathbf{x}_2 \\ \mathbf{M} \\ \Delta \mathbf{x}_q \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \mathbf{M} \\ \mathbf{S}_q \end{bmatrix} \quad (q=1, n),$$

где  $\Delta \mathbf{x}_i$ ,  $\mathbf{S}_i$  ( $i=1, q$ ) имеют тот же смысл, что и (2), (7).

**2. Результаты расчетных исследований.** Параметры системы определяются элементами матриц  $\mathbf{A}$  и  $\mathbf{B}$ :

$$\mathbf{A} = \begin{bmatrix} 1,643 & 0 & -0,053 & 0 & -0,091 & 0 & -0,013 \\ 0 & 1,643 & 0,053 & 0,091 & 0 & 0 & -0,013 \\ -0,053 & 0,053 & 0,015 & 0 & 0 & 0 & 0 \\ 0 & 0,091 & 0 & 0,017 & 0 & 0 & 0 \\ -0,091 & 0 & 0 & 0 & 0,017 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,534 & 0 \\ -0,013 & -0,013 & 0 & 0 & 0 & 0 & 7,97 \cdot 10^{-3} \end{bmatrix};$$

$$B = \text{diag}\{5,809 \cdot 10^3 \quad 5,809 \cdot 10^3 \quad 21,413 \quad 97,469 \quad 97,469 \quad 2,213 \cdot 10^5 \quad 174,7\}$$

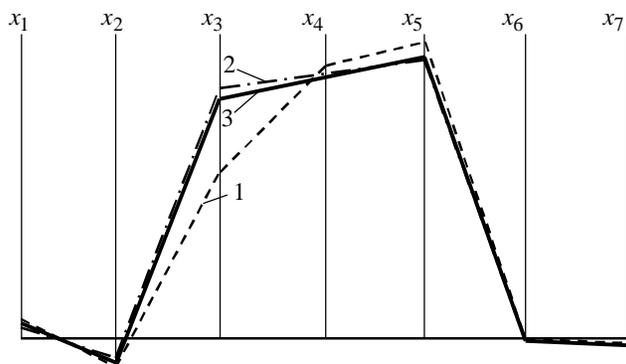


Рис. 1. Результаты оптимизации собственного вектора: 1 – исходное значение вектора; 2 – желаемое значение; 3 – значение после оптимизации

На рис. 1 представлены результаты оптимизации пятого собственного вектора с помощью двух параметров. Исходные значения указанных параметров  $h = 0,14 \cdot 10^{-2}$ ;  $d = 0,2 \cdot 10^{-2}$ . Ломаными линиями показаны исходный собственный вектор, его желаемая конфигурация и результаты оптимизации. В качестве желаемой конфигурации собственного брался исходный вектор, у которого значение координаты  $x_3$  увеличено на 30% и настолько же процентов уменьшены значения координат  $x_1$  и  $x_2$ . Значения варьируемых параметров при этом оказались следующими:  $h = 0,177 \cdot 10^{-2}$ ;  $d = 0,288 \cdot 10^{-2}$ .

**Выводы.** 1. Дан анализ современного состояния применения математических методов и моделей в стратегическом планировании и управлении проектами. 2. Предложен математический аппарат оптимизации внутренних экономических показателей как решение задачи синтеза собственных векторов соответствующих матриц. 3. Приведены результаты расчетов по оптимизации параметров матрицы за счет синтеза координат собственного вектора.

**Список литературы:** 1. Оптимальное управление экономическими системами / Под ред. П.Д. Шимко. – СПб: Бизнес-пресса, 2004. – 240с. 2. Игнатьева А.В., Максимцов М.М. Исследование систем управления. – М.:ЮНИТИ-ДАНА, 2000. – 157с. 3. Charles W. White and Bruce D. Maytum. Eigensolution sensitivity to parametric model perturbations // The shock and vibration bulletin.–1976.–N. 46.–P. 123-133.

Поступила в редакцию 07.12.09

УДК 338.2

**М. Д. ГОДЛЕВСКИЙ**, д-р техн. наук, проф., зав. каф. АСУ НТУ «ХПИ»,  
**А. А. СТАНКЕВИЧ**, аспирант каф. АСУ НТУ «ХПИ»

### КЛАССИФИКАЦИЯ ИЕРАРХИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ И КООРДИНАЦИИ БИЗНЕС-ПРОЦЕССОВ ЦЕПОЧЕК ПОСТАВОК

У роботі зроблено класифікацію ієрархічних систем управління і координації бізнес-процесів логістичних ланцюгів поставок. Розглянуто двох і трьохрівневих ієрархічних системи. Запропоновано новий тип фірм – регіональних системних інтеграторів бізнес-процесів.

В работе сделана классификация иерархических систем управления и координации бизнес-процессов логистических цепочек поставок. Рассмотрены двух и трехуровневые иерархические системы. Предложен новый тип фирм – региональных системных интеграторов бизнес-процессов.

In this work classification of hierarchical control systems and coordination of business processes of logistical chains of deliveries is made. Two and three-level hierarchical systems are considered. The new type of firms – regional system integrators of business processes is offered.

**Введение.** С конца 1980-х годов все активнее стал употребляться термин «управление цепочками поставок» (supply chain management, SCM). Фактически SCM – это интегрирование ключевых бизнес-процессов, начинающихся от конечного пользователя и охватывающих всех поставщиков: сырья, материалов, готовой продукции, различных услуг и информации, добавляющих ценность для потребителей и других заинтересованных лиц [1]. С другой стороны, SCM представляет взаимодействие трех элементов: структуры цепочек поставок; бизнес-процессов, происходящих в цепочках поставок; компонентов управления цепочками поставок.

Можно выделить три составляющие всех цепочек: снабжение (сырье и материалы); незавершенное производство; реализация готовой продукции. Основной задачей управления цепочками поставок является оптимизация не отдельных бизнес-процессов (транспортные задачи, управление запасами и т.д.), а их интеграция и координация, что позволяет значительно уменьшить суммарные затраты [2].

**Двухуровневые системы управления.** Проведем классификацию иерархических систем управления и координации бизнес-процессов цепочек поставок независимо от того, какая составляющая логистической цепи поставок рассматривается. В дальнейшем для основного предприятия (центральной компании), которое производит оригинальную продукцию, будем использовать аббревиатуру OEM (Original Equipment Manufacturer). Цепочки поставок имеют три структурных размерности: горизонтальная

структура; вертикальная структура и горизонтальное положение OEM. В тех случаях, когда число горизонтальных уровней не велико и количество поставщиков/потребителей в каждом уровне также не велико («узкая» вертикальная структура), то иерархическая система управления цепочками поставок может иметь следующий вид (рис. 1).  $BP_i$  – отдельный бизнес-процесс логистической цепочки поставок.  $I$  – множество бизнес-процессов, которыми непосредственно управляют менеджеры OEM,  $i \in I$ .

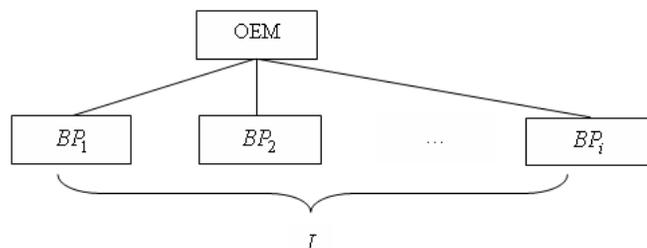


Рис. 1. Простейшая двухуровневая структура системы управления цепочками поставок

По мере увеличения числа горизонтальных уровней цепи поставок, а также количества поставщиков/потребителей продукции, на каждом уровне все больше возрастает напряженность в использовании ресурсов OEM и ограничивается число процессов, которые центральная компания может интегрировать за пределами первого уровня и которыми она может реально управлять. Фактически менеджеры OEM в этом случае в состоянии активно управлять только ограниченным количеством бизнес-процессов. Поэтому они используют аутсорсинг, передавая выполнение отдельных бизнес-процессов компаниям, которые в зарубежной литературе называются Contract Logistics Service Provider. В дальнейшем для них будем использовать следующую аббревиатуру 3PL провайдер (Third Party Logistics Provider). В этом случае структура двухуровневой иерархической системы управления цепочками поставок приведена на рис. 2.  $3PL_j$  – отдельный  $j$ -й провайдер.  $J$  – множество 3PL провайдеров, непосредственно управляемых и координируемых OEM компанией,  $j \in J$ .

Базовыми функциями для 3PL провайдеров являются: транспортные перевозки, складирование, управление запасами, обслуживание покупателей, информационные услуги. Однако в настоящее время они берут на себя новые функции такие как: управление возвратными потоками, планирование выпуска продукции, выбор перевозчиков и т.д. Если раньше 3PL провайдеры

ориентировались на услуги, которые требовали потребители, то в настоящее время передовые 3PL провайдеры активно внедряют на рынке новые области обслуживания, предлагая потребителям инновационные решения. Необходимо отметить, что 3PL провайдеры отличаются друг от друга не только по типам бизнес-процессов, которые они выполняют, но и по отсутствию или наличию активов, которыми они обладают. Например, наличие собственных транспортных средств для осуществления перевозок или их аренда. Естественно, что возможны некоторые промежуточные варианты, когда осуществляется частичная аренда активов.

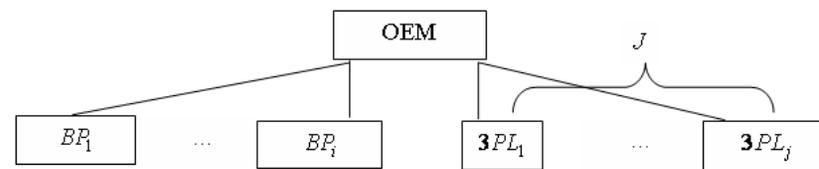


Рис. 2. Двухуровневая структура системы управления цепочками поставок

**Трехуровневые системы управления.** Для транснациональных компаний, имеющих множество рынков сбыта продукции на различных континентах, которые сырье и материалы получают на основе множества уровней поставщиков, возникает проблема отсутствия ресурсов, а также экономической целесообразности для координации функционирования большого количества 3PL провайдеров. Поэтому на новом этапе развития логистических услуг появился новый тип провайдеров, которые были названы системными интеграторами (System Integrator) или 4PL провайдерами (Fourth Party Logistics Provider). Фактически, они берут на себя обязанности OEM компании по координации функционирования отдельных логистических цепочек поставок. Основной отличительной особенностью 4PL провайдеров является то, что они не имеют своих собственных активов и не решают свои отдельные локальные задачи с точки зрения идеологии иерархических систем управления. Основная их задача – координация (интеграция) бизнес-процессов, которые выполняют 3PL провайдеры с точки зрения интересов OEM компании. В этом случае мы имеем трехуровневую иерархическую систему управления цепочками поставок, которая приведена на рис. 3.  $K$  – множество 4PL провайдеров, которые непосредственно связаны с OEM компанией,  $\bar{J}_k$  – множество  $3PL_{kj}^n$  провайдеров, координируемых  $k$ -м 4PL системным интегратором, где  $j \in \bar{J}_k$ ,  $k \in K$ .

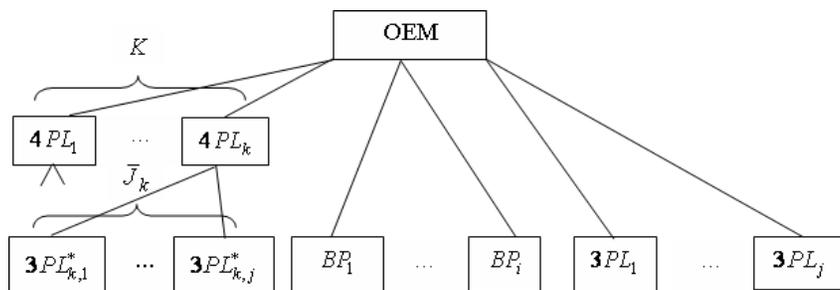


Рис. 3. Трехуровневая структура системы управления цепочками поставок

Дальнейшее развитие логистических услуг связано с появлением нового типа компаний, которые выполняют функции, аналогичные 4PL системным интеграторам по координации бизнес-процессов, выполняемых 3PL провайдерами, но в отличие от 4PL компаний они имеют свои активы и самостоятельно выполняют отдельные бизнес-процессы в цепочке поставок. Такие компании в зарубежной литературе имеют аббревиатуру LLP (Lead Logistics Provider). Привлекательность таких компаний для OEM связана с тем, что они берут полностью на себя ответственность за выполнение и координацию ряда бизнес-процессов цепочек поставок, высвобождая ресурс OEM для решения более важных и собственных ей задач. Таким образом, трехуровневая структура системы управления цепочками поставок становится еще более сложной и имеет следующий вид (рис. 4).  $L$  – множество LLP фирм, которые непосредственно связаны с OEM,  $\bar{I}_l$  – множество бизнес-процессов  $BP_{li}^*$ , которые выполняются и координируются  $l$ -й LLP фирмой, где  $i \in \bar{I}_l$ ,  $l \in L$ .

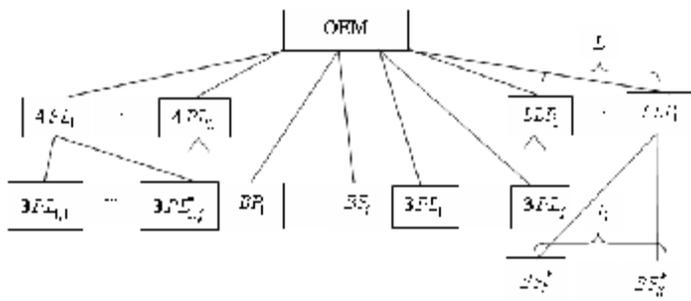


Рис. 4. Трехуровневая структура системы управления цепочками поставок с использованием LLP фирм

Выше приведены структуры иерархических систем управления, относящихся к цепочкам поставок отдельных центральных компаний. На определенной территории (в некотором регионе) обычно функционируют различные производители оригинальной продукции. Таким образом, можно говорить о логистических цепочках поставок различных OEM. Интеграция таких цепочек позволит уменьшить суммарные расходы заинтересованных в этом процессе OEM. Поэтому авторы работы предлагают рассмотреть возможность функционирования двух новых типов компаний.

1. Компании, которые в дальнейшем будем называть региональными системными интеграторами с аббревиатурой RSI (Regional System Integrator). Такие компании являются дальнейшим развитием 4PL провайдеров. Они не имеют своих собственных активов и по договору с рядом OEM компаний, функционирующих в некотором регионе, занимаются вопросом координации функционирования 3PL провайдеров, работающих с различными OEM фирмами.

2. Компании, которые в дальнейшем будем называть региональными LLP фирмами с аббревиатурой RLLP (Regional Lead Logistics Provider). Фактически, они являются дальнейшим развитием LLP фирм, имеют свои собственные активы и по договору с рядом OEM компаний решают вопросы реализации и координации функционирования бизнес-процессов цепочек поставок различных OEM фирм в пределах некоторого региона. Частично они могут по договору с OEM фирмами передать выполнение отдельных бизнес-процессов 3PL провайдерам.

В обоих этих случаях структура иерархической системы управления и координации цепочек поставок является более сложной по сравнению с выше приведенными с возможностью реализации различных ее вариантов функционирования в зависимости от конкретных ситуаций: размеров региона, количества OEM, вида товаров и услуг и т.д. Например, с горизонтальной координацией (по уровням цепочек поставок) или с вертикальной координацией цепочек поставок различных OEM. Очевидно, в некоторых случаях может быть целесообразным итерационный процесс управления с вертикальной и горизонтальной координацией бизнес-процессов цепочек поставок различных OEM. Рассмотрим отдельные структуры иерархических систем управления цепочками поставок с участием RSI и RLLP фирм.

Допустим, в некотором регионе функционирует множество  $S$  OEM компаний. В этом случае мы имеем трехуровневую структуру системы управления цепочками поставок различных OEM компаний, которая приведена на рис. 5.  $OEM_s$  –  $s$ -я центральная компания, где  $s \in S$ ;  $\tilde{J}^s$  – множество  $3PL_j^s$ -х провайдеров, выполняющих бизнес-процессы цепочки

поставок OEM<sub>S</sub>-ой фирмы и координируемых RSI фирмой. LLP<sub>i</sub> и 4PL<sub>k</sub> – фирмы, работающие вне рассматриваемого региона.

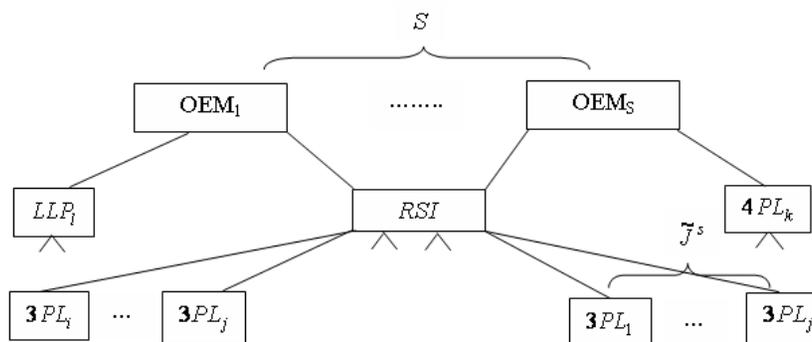


Рис. 5. Трехуровневая структура системы управления цепочками поставок в регионе на основе RSI-фирмы

**Заключение.** В работе рассмотрены различные типы фирм, участвующих в реализации бизнес-процессов цепочек поставок. Предложена классификация иерархических систем управления и координации бизнес-процессов логистических цепочек поставок. В зависимости от назначения центральной фирмы, длины горизонтальной структуры цепочек поставок, «ширины» вертикальной структуры и горизонтального положения OEM фирмы рассмотрены различные виды двух и трехуровневых иерархических систем управления. Впервые предложены два новых типа фирм – региональных системных интеграторов, основное отличие которых друг от друга заключается в отсутствии и наличии активов. Дальнейшие исследования в этом направлении связаны с разработкой моделей и алгоритмов координации бизнес-процессов в рамках двух и трехуровневых иерархических систем управления. В зависимости от типа фирм, участвующих в координации бизнес-процессов, а также вида структуры иерархической системы управления будут рассмотрены различные механизмы управления.

**Список литературы:** 1. Сток Дж Р., Ламберт Д. М. Стратегическое управление логистикой. – М.: ИНФРА-М, 2005. – 797 с. 2. Годлевский М. Д., Станкевич А. А. Принципы структурно-параметрического синтеза модели транспортно-складской системы транснациональной логистической компании. – Вестник НТУ «ХПИ». Сб. научных трудов. Тем. вып. «Системный анализ, управление и информационные технологии». – Харьков: НТУ «ХПИ». – 2009. – № 10. – С. 23-30.

Поступила в редколлегию 17.12.09

УДК 519.816

**М. Д. ГОДЛЕВСКИЙ**, д-р техн. наук, проф., зав. каф. АСУ НТУ «ХПИ»,  
**В. Ю. ВОЛОВЩИКОВ**, канд. техн. наук, доц. каф. АСУ НТУ «ХПИ»,  
**В. Ф. ШАПО**, канд. техн. наук, доц. каф. ТАУиВТ ОНМА (г. Одесса)

### ПРИНЦИПЫ ПОСТРОЕНИЯ ТРАЕКТОРИИ РАЗВИТИЯ КОРПОРАТИВНОЙ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ СЛУЧАЯ НЕНАСЛЕДУЕМЫХ СТРУКТУР

В статті розглядаються проблеми використання принципу успадкованості структур у задачі розвитку корпоративної інформаційно-обчислювальної системи (КІОС). Розроблено підхід до формування раціональної траєкторії розвитку КІОС на основі принципу неуспадкованості.

В статье рассматриваются проблемы использования принципа наследуемости структур в задаче развития корпоративной информационно-вычислительной системы (КИВС). Разработан подход к формированию рациональной траектории развития КИВС на основе принципа ненаследуемости.

In article problems of use of a principle of heritability of structures in a task of development of enterprise information system (EIS) are considered. The approach to formation of a rational trajectory of development EIS is developed on the basis of a nonheritable's principle.

**Введение.** Сегодня достаточно трудно представить эффективно работающее предприятие, которое не применяло бы у себя современные информационные технологии. Успех использования информационных технологий могут обеспечивать КИВС. К важным особенностям КИВС, с точки зрения данной работы, можно отнести их постоянное совершенствование и развитие. Это в свою очередь обеспечивает повышение эффективности работы персонала, улучшение качества принимаемых управленческих решений, достижение поставленных целей и т.д. Однако все это может быть достигнуто только за счет грамотного решения задачи развития КИВС.

Классическая теория [1], рассматривающая динамическую задачу структурного синтеза КИВС, исходит из принципа наследуемости. При этом предполагается, что на каждом  $(t+1)$ -м этапе синтезируемая структура КИВС должна обладать свойством вложения по отношению к структуре  $t$ -го этапа. То есть, на каждом  $(t+1)$ -м этапе структура КИВС  $D_{t+1}$  получается путем расширения структуры  $D_t$  предыдущего  $t$ -го этапа. При этом введенные ранее элементы КИВС не удаляются, а используются на всех последующих этапах. Таким образом, искомая последовательность структур будет обладать свойством вложения:  $D_1 \subseteq \dots \subseteq D_t \subseteq D_{t+1} \subseteq \dots \subseteq D_T$ .

Использование принципа наследуемости не допускает удаления (консервирования) существующих элементов КИВС, что, с точки зрения авторов данной работы, может не обеспечивать выбор оптимального решения за счет отсутствия возможности рационального развития КИВС при

существенно сокращенном множестве вариантов альтернатив. В связи с этим к решению задачи развития КИВС предлагается применять новый подход, который позволял бы разрешить эту проблему.

**Постановка задачи.** Рассмотрим КИВС, основной задачей которой является удовлетворение потребностей в выполнении информационно-вычислительных работ предприятия на плановом периоде  $[1, T]$ . Для оценки эффективности каждого варианта синтезируемых структур КИВС для каждого этапа планирования рассматриваются показатели, введенные в [2].

Особенностью таких критериев эффективности КИВС, как минимизация среднего времени задержки и максимизация структурной живучести, является то, что они должны характеризовать качество синтезируемой структуры КИВС в каждый отдельный этап планового периода. При этом излишек или недостаток численных значений этих показателей не может быть компенсирован на последующих этапах. Однако показатель затрат на организацию и эксплуатацию КИВС при решении данной задачи может быть выражен как интегральный показатель на всем плановом периоде  $[1, T]$ .

Тогда для решения задачи развития КИВС предлагается использовать иной подход по отношению к принципу преемственности [1]. В основу нового подхода положим принцип ненаследуемости. В этом случае будем оперировать тремя базовыми свойствами: вложение, частичное вложение и полная невложенность. Если структура КИВС  $(t+1)$ -го этапа может быть получена путем расширения структуры  $t$ -го этапа, будем говорить о соблюдении свойства вложения. Если при переходе от  $t$ -го к  $(t+1)$ -му этапу использование части элементов КИВС нецелесообразно, соседние структуры будут обладать свойством частичного вложения. Теоретически возможен вариант, когда пересечение структур двух соседних этапов может быть пустым, тогда будем говорить о тождественности свойства полной невложенности. В случае проявления двух последних свойств будем говорить, что неиспользуемые элементы КИВС консервируются. То есть временно приостанавливается их использование.

Таким образом, целью работы является разработка основных принципов построения траектории развития КИВС, которая определяла бы рациональный вариант ее развития при обозначенных выше критериях для случая ненаследуемых структур.

**Методика построения траектории развития КИВС.** Так как в основу построения рациональной траектории развития КИВС предлагается положить принцип ненаследуемости, то на первом шаге логично сформировать множество допустимых вариантов структур  $\{u_p^t, p \in \Pi^t\}$  для каждого  $t$ -го этапа планирования путем решения ряда статических задач структурного синтеза КИВС [2] с использованием метода главного критерия [3] в качестве которого выступает критерий затрат на организацию и эксплуатацию КИВС.

На втором шаге для каждой пары опорных состояний  $(u_p^{t-1}, u_{\bar{p}}^t)$ ,  $p \in \Pi^{t-1}$ ,  $\bar{p} \in \Pi^t$  необходимо оценить возможность проявления одного из выше введенных свойств: вложение, частичное вложение или полная невложенность. Так как в общем случае переход от структуры  $(t-1)$ -го к структуре КИВС  $t$ -го этапа может быть выполнен одним из допустимых вариантов  $card(\Pi^{t-1}) \cdot card(\Pi^t)$ , то относительно каждого  $(p, \bar{p})$ -го перехода может наблюдаться одна из следующих ситуаций.

Во-первых,  $p$ -структура КИВС  $(t-1)$ -го этапа, определяемая опорным состоянием  $u_p^{t-1}$ , может быть полностью вложена в  $\bar{p}$ -структуру КИВС  $t$ -го этапа с опорным состоянием  $u_{\bar{p}}^t$ . Во-вторых, структура  $u_p^{t-1}$  может быть частично вложена по отношению к структуре  $u_{\bar{p}}^t$ , то есть, можно выделить подструктуру  $d_p^{t-1}$  структуры  $u_p^{t-1}$ , элементы которой наследуются в  $u_{\bar{p}}^t$ . В тоже время элементы подструктуры  $u_p^{t-1} \setminus d_p^{t-1}$  будут считаться законсервированными для  $\bar{p}$ -структуры  $t$ -го этапа. В-третьих, структуры  $u_p^{t-1}$  и  $u_{\bar{p}}^t$  могут не иметь общих элементов, то есть  $u_p^{t-1} \cap u_{\bar{p}}^t = \emptyset$ , что определено как полная невложенность (все элементы КИВС структуры  $u_p^{t-1}$  в  $\bar{p}$ -й структуре  $t$ -го этапа будут законсервированы).

Использование принципа ненаследуемых структур в задаче управления развитием КИВС предполагает необходимость выполнения специальной процедуры для каждого  $(p, \bar{p})$ -го перехода независимо от того, какое наблюдается свойство (вложение, частичное вложение и полная невложенность) для  $u_p^{t-1}$  и  $u_{\bar{p}}^t$  структур. Необходимость дополнительной процедуры заключается в том, что законсервированные элементы предыдущих этапов могут использоваться на последующих, что оказывает непосредственное влияние на размер затрат каждого следующего этапа.

Рассмотрим процедуру коррекции размера затрат на организацию и эксплуатацию КИВС.

Согласно принципу ненаследуемых структур, в общем случае,  $u_p^t$  опорное состояние может быть оптимальным опорным состоянием рациональной траектории развития КИВС. При этом общее число допустимых путей, которые могут обеспечить переход в  $u_p^t$  состояние, определится величиной  $\prod_{i=1}^{t-1} card(\Pi^i)$ ,  $t = [2, T]$ . Тогда каждому  $j$ -му пути в соответствие можно поставить три множества.

Пусть первое множество  $\Omega_j$  определяет номера опорных состояний, образующих  $j$ -й путь.

Следовательно вторым множеством  $\bigcup_{p \in \Omega_j, t=1, t-1}^i d_p^i$  можно описать множество унаследованных элементов для  $p$ -го опорного состояния  $t$ -го этапа. Тогда, третьим множеством  $\bigcup_{p \in \Omega_j, t=1, t-1}^i (u_p^i \setminus d_p^i)$  определим множество законсервированных

элементов КИВС относительно  $u_p^i$  опорного состояния.

Таким образом, множество элементов КИВС, определяемое как  $u_p^i \setminus \left( \left( \bigcup_{p \in \Omega_j, t=1, t-1}^i d_p^i \right) \cup \left( \bigcup_{p \in \Omega_j, t=1, t-1}^i (u_p^i \setminus d_p^i) \right) \right)$ , характеризует такие элементы  $u_p^i$  опорного состояния при переходе в него  $j$ -м путем, которые являются ни унаследованными, ни ранее законсервированными. Именно такие элементы определяют реальные затраты на организацию и эксплуатацию КИВС для  $p$ -го опорного состояния  $t$ -го этапа планирования. Тогда оценка  $u_p^i$  опорного состояния с точки зрения затрат определится выражением:

$$\frac{\min_{j=1, \prod_{t=1}^{t-1} card(\Pi^i)} F_c^{u_p^i}}{\left( u_p^i \setminus \left( \left( \bigcup_{p \in \Omega_j, t=1, t-1}^i d_p^i \right) \cup \left( \bigcup_{p \in \Omega_j, t=1, t-1}^i (u_p^i \setminus d_p^i) \right) \right) \right)}$$

Следует заметить, что каждое  $u_p^i$  опорное состояние определяется тройкой значений  $F_c^{u_p^i}$ ,  $F_t^{u_p^i}$  и  $F_p^{u_p^i}$ , соответственно характеризующих реальные затраты на организацию и эксплуатацию, оперативность и живучесть  $u_p^i$  опорного состояния КИВС.

На третьем шаге формирования рациональной траектории развития КИВС на основе принципа ненаследуемых структур определим следующее допущение.

Так как с одной стороны каждое  $u_p^i$  опорное состояние определяется  $F_c^{u_p^i}$ ,  $F_t^{u_p^i}$  и  $F_p^{u_p^i}$ , причем  $F_t^{u_p^i}$  и  $F_p^{u_p^i}$  являются допустимыми, а с другой стороны затраты на организацию и эксплуатацию КИВС должны быть выражены в виде интегрального показателя на всем плановом периоде  $[1, T]$ ,

то при построении траектории развития КИВС будем учитывать только характеристику  $F_c^{u_p^i}$ .

С целью формирования обобщенной оценки каждого варианта траектории развития КИВС, общее количество которых определяется  $\prod_{t=1}^i card(\Pi^i)$ ,  $t = [2, T]$ , на четвертом шаге алгоритма построим соответствующую интегральную оценку. Причем с каждым  $j$ -м допустимым путем развития КИВС свяжем определенную реализацию интегральной оценки.

Тогда интегральную оценку  $j$ -го пути определим, как:

$$F^j = \sum_{p \in \Pi_j^i, t \in [1, T]} F_c^{u_p^i}, j = 1, \prod_{t=1}^{t-1} card(\Pi^i), \quad (1)$$

где  $\Pi_j^i$  – номер опорного состояния КИВС на  $t$ -м этапе планирования, находящийся на  $j$ -м пути.

Следовательно, путь  $j'$ , которому можно поставить в соответствие наименьшее значение интегральной оценки (1):

$$F^{j'} = \min_{j=1, \prod_{t=1}^{t-1} card(\Pi^i), t \in [2, T]} \{ F^j \}$$

определил номера опорных состояний, соответствующих рациональной траектории развития КИВС на периоде планирования  $[1, T]$ .

**Выводы.** В данной научной работе сформулирована проблема рационального развития КИВС с использованием классического подхода, основанного на принципе наследуемых структур. Введены понятия, связанные с новым подходом к развитию КИВС, базирующиеся на введенном принципе ненаследуемых структур. Разработана методика построения рациональной траектории развития КИВС для случая ненаследуемых структур.

**Список литературы:** 1. Зайченко Ю. П., Гонга Ю. В. Структурная оптимизация сетей ЭВМ – К.: Техника, 1986. – 168 с. 2. Годлевский М. Д., Воловщиков В. Ю. Модель статической задачи структурного синтеза корпоративной информационно-вычислительной системы // Восточно-Европейский журнал передовых технологий. - 2006. - № 2. - С. 110-113. 3. Подиновский В. В., Ногин В. Д. Парето-оптимальные решения многокритериальных задач. - М.: Наука, 1982. - 256 с.

Поступила в редколлегию 20.11.09

V. A. *SHEKHOVTSOV*, PhD, NTU «KhPI»

## TOWARDS ASSESSING SIMULATED SERVICE QUALITIES BY BUSINESS STAKEHOLDERS OUTSIDE THE SYSTEM BOUNDARY

У роботі описана технологія збору думок реальних зацікавлених осіб (які не взаємодіють із системою на її границі) про якість сервіс-орієнтованих програмних систем. Ця технологія є розширенням методу ISAREAD-S, що реалізує збір таких думок на основі оцінювання характеристик якості програмних сервісів, поданих імітаційними моделями, на границі системи в контексті процесів їх використання. Це розширення полягає у реалізації спеціальної моделі залежності сервісів, що враховує залежності між показниками якості діяльності процесів, доступних для реальних зацікавлених осіб і показниками якості діяльності, визначених на границі системи, які представляють сервіси, що розроблюються.

В работе описана технология сбора мнений реальных заинтересованных лиц (которые не взаимодействуют с системой на ее границе) о качестве сервис-ориентированных программных систем. Данная технология является расширением метода ISAREAD-S, реализующего сбор таких мнений на основе оценивания характеристик качества программных сервисов, представленных имитационными моделями, на границе системы в контексте процессов их использования. Это расширение сводится к реализации специальной модели зависимости сервисов, учитывающей зависимости между показателями качества деятельности процессов, доступных для реальных заинтересованных лиц и показателями качества деятельности, определенных на границе системы, которые представляют разрабатываемые сервисы.

The paper describes an approach for collecting opinions on quality of service-oriented systems from the specific category of real business stakeholders (which do not interact with such systems directly on their boundary). This approach enhances the ISAREAD-S method for collecting assessments of simulated service qualities on the system boundary in context defined by the service usage process. We introduce special Service Dependency Model taking into account dependencies between usage process activities accessible by real stakeholders and boundary activities representing services under development.

**1. Introduction.** Software development process for service-oriented systems often relies on the assumption that every service under development can be directly assessed by stakeholders, i.e. every stakeholder is a *direct user* of the system under development (SUD) treated as a set of services. This is mostly true for the systems where all the service users are at the same time the business customers (i.e. all interactions with services happen at the system boundary) such as e-Business systems (online shops, auctions etc). However, for many systems this is not the case; to reflect this [5, 6] draw a line between such direct users and *real users* (customers of the system).

For example, in the system installed at the hotel reception desk, direct users are hotel receptionists whereas real users are its customers. Real users participate in system usage processes (e.g., the process of reserving a room) but do not interact directly with the system at its boundary (this is a work for the receptionists). Ignoring such users can lead to the biased treatment of the prospective system, missing important requirements, customer dissatisfaction, and the project failures.

**Paper context.** To facilitate development of the service-oriented systems, we proposed the ISAREAD-S framework (Interactive Simulation-Aided Requirements Engineering and Architectural Design for Services) [9, 13, 14] aimed at investigating the ways to support the stakeholder involvement in a form of assessing the perceived quality of the service-oriented SUD in its usage context. To implement such support we plan to elaborate a simulation-based method of making service quality assessment procedures accessible to the business stakeholders (without background in IT) and using their assessments as a driving force for software process activities related to requirements engineering and architectural design. This paper is devoted to integrating real users' assessments into this framework; it describes service-level mechanisms for assessing the system qualities outside its boundary and their integration into process-level mechanisms.

**Paper structure.** Section 2 describes the state of the art and formulates the problem statement, Section 3 shows the principles of the existing mechanisms for organizing the interaction with stakeholders on the system boundary, Section 4 outlines the proposed solution introducing specific service-level mechanisms for assessing derived service qualities outside the system boundary, Section 5 makes conclusions and describes the directions for future research.

**2. State of the art and problem statement.** To classify the existing methods for stakeholder involvement in the software process we follow the taxonomy of the methods to represent the quality of the prospective system proposed by Bosch [2]; it includes scenario-based techniques, prototyping, and simulation. In this paper, we restrict ourselves to methods belonging to the first category, see [14] for the review of methods belonging to other categories.

**Human interaction with stakeholders.** These methods directly address the problem of involving stakeholders into the software process by performing human interaction with them and collecting their opinions. We describe scenario-centered techniques of such involvement [3, 4, 7, 8, 11, 12] which organize scenarios of stakeholder interaction with a prospective system. In most cases stakeholders are requested to proceed through the scenarios in manual mode and express their opinions. There are shortcomings of these techniques as a means of addressing our problem: stakeholders cannot experience quality in a way that they perceive it as real; it is not possible to investigate the dependency between quality-influencing factors and the observed quality levels [2]; for reliability, the scenarios mostly cannot replace interaction with the real system or its executable model as they are able to express it only by example [1].

**Real users and their involvement into the software process.** The concept of real users which interact with the system outside its boundary was first introduced in [5]. In this book, these users were called customers, whereas direct users which interact with the system on its boundary were called participants. More extended treatment of this category of users was proposed in [6]. In this work the problem of gathering requirements from these users was investigated in detail. No attempts,

however, were made to make such customers experience qualities of the future system and assess these qualities in a way that these assessments become usable in a role of the software process' driving forces.

**Problem statement.** After analyzing the state of the art we can formulate both general and specific research questions which determine the problem statement.

The general question is: *How to involve business stakeholders into the development process for service-oriented software systems as a means of control for the performance and reliability of the produced artifacts?* We address this question by introducing ISAREAD-S framework [14] offering mechanisms for interactive assessment of simulated service performance and reliability on the system boundary; we present an outline of this framework in the next section.

The specific research question related to the topic of this paper is: *How to involve real users of the service-oriented system into the quality assessment process?* To answer this question, it is necessary to establish the necessary models and procedures which would allow qualities of the specific activities positioned outside the system boundary to be derived from the qualities of the services under development and to be proposed to the business stakeholders for assessment.

**3. Assessment mechanisms on the system boundary.** In [14] we described the proposed approach to establish service-level and process-level assessment mechanisms for the case when the services are directly accessible to stakeholders.

**Service-level mechanisms.** We elaborated IAS mechanisms (short for *Interactive Assessment of Services*) aimed at an assessment of simulated service qualities at the level of the particular service. According to the model-driven methodology [10] it is necessary to have two mechanisms of this kind: IASC (for model composition) and IASE (for model execution). IASC inputs include the set of qualities of interest to be simulated and assessed and the set of factors influencing the simulation (simulation parameters [14]). To get the integrated quality simulation model, we compose simulation modules corresponding to the qualities of interest and the necessary parameters together with the base simulation structure. Also, we integrate into this model the set of user interaction models for the qualities of interest. The resulting service-level simulation and assessment model becomes the IASC output. It is transferred to IASE for standalone execution.

IASE is responsible for execution of both simulation and assessment interaction submodels of IASM. The input for every IASE run is the set of parameter values corresponding to the parameters used to build IASM. As a result of the run, the set of simulated values for the qualities of interest is obtained and presented to the service user for assessment via interaction processes described by interaction models integrated into IASM. The IASE outputs are this set of simulated qualities and the set of assessment results.

**Process-level mechanisms.** We elaborated IAP mechanisms (short for *Interactive Assessment of Processes*) aiming at interactive assessment of simulated service qualities in context of usage processes at the level of the particular process,

in particular: IAPC (for model composition) and IAPE (for model execution). They rely on IASC and IASE dealing with individual services.

IAPC forms the simulation model of the usage process making it ready for interactive assessment of service qualities. It combines the control flow model (CFM) for the usage process (conforming to the network BPM notation) with the role model for the usage process. The role model includes the set of roles defined for process participants (clerk, manager etc), the sets of interaction activities for different roles (they make participants affect the state of the process simulation), the sets of assessment activities for different roles (they correspond to the services of interest to be simulated and assessed by stakeholders) and the sets of qualities of interest and necessary parameters defined for every service of interest.

While composing the integrated model IAPM for the process, IASC creates the IASM model for every service of interest; this model later becomes integrated into IAPM. For every interaction activity, a mechanism for constructing the interaction model is invoked and the resulting interaction model is also integrated into IAPM. The resulting model will contain the simulation logic defined by CFM for the process and simulation submodels of different IASM models (for the services of interest); the assessment logic defined by interaction submodels of these IASM models; the interaction logic defined for all interaction activities.

The IAPM is executed by IAPE. Every run is presented to the stakeholder belonging to the particular role. During the run, the basic simulation flow is managed by the model derived from the CFM of the usage process; when the logic of the run requires invoking an activity representing the service of interest, the simulation of its qualities and the assessment interaction logic are handled by IASE invoked for its IASM. IASE inputs are parameter values for all the slots of this service; when this logic requires interacting with the simulation, the logic of this interaction is handled by the corresponding interaction mechanism. The outputs for IAPE run include the set of all simulated quality values for all the services of interest and the set of corresponding assessment results.

**4. Outline of the proposed solution.** We plan to address the above research question by extending the process-level mechanisms taking into account the dependencies between activities accessible by the real users and boundary activities representing services under development. We introduce two additional sets of business process activities to be defined for every role:

1. The set of derived activities accessible to the real users which could assess their qualities. They lie outside the system boundary and do not represent services so their qualities are not simulated directly;
2. The set of hidden activities representing services not accessible to the real users. Their qualities are simulated but not shown to stakeholders; they are used to derive the "visible" qualities of the derived activities. Such activity can be at the same time an assessment activity for the other role representing its direct user.

To take real users into account, we plan to introduce new service-level IAD mechanisms (interactive assessments of derived values) for handling real user interactions with the derived activities (IADC and IADE).

The additional input for IADC is the service dependency model SDM for a (derived) activity. It describes the dependencies between “visible” qualities of this activity and the simulated qualities of the services it depends on (its base services). Taking SDM into account (in addition to qualities of interest), IADC prepares the IADM model for an activity which describes how the presented values can be obtained from the simulated values. IADC gets the set of IAM from IAML and integrates them with the dependency model. To simplify SDM implementation, we plan to establish a library of basic dependency rules.

Executing IADM with IADE is shown on Fig.1a. IADE obtains the simulated values for all the qualities of the activity’s base services. It then makes forward transformation of these values into the quality values for an activity. These values are shown to the real user to assess. In the simplest case the assessments obtained from this user form the output of IADE together with the simulated qualities of base services and the derived qualities and can be used to help in making informed decisions by analysts.

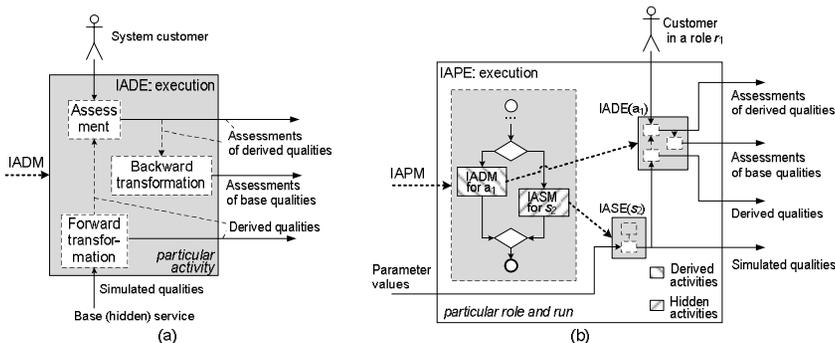


Fig.1. Service-level assessment by real users (a); its integration into the process-level mechanism (b)

In the “ideal” case, IADE makes backward transformation of such stakeholder assessments into the assessments of the base services’ qualities. This way, IADE becomes output-compatible with IASE. This transformation is easier to implement if the activity has a single base service. If there are multiple base services, the situation is more complicated (we expect that varying input parameter values and, as a result, simulated qualities for different base services and observing resulting variations in real user assessments could help in such situations). We plan to investigate this issue in detail to find the dependency configurations making possible such transformations.

IADC and IADE are integrated into process-level mechanisms similarly to IASC and IASE. The execution of simulation model by IAPE with boundary and derived activities is shown on Fig.1b. For the “hidden” base service  $s_2$ , IASE is invoked with disabled interaction subsystem, its simulated quality values become an input for the forward transformation module of IADE ( $a_1$ ).

**5. Conclusions and future research directions.** In this paper, we defined new service-level mechanisms for assessing the performance and reliability of the services under development by business stakeholders which take into account the dependencies between activities accessible by the real users and boundary activities representing services under development. These mechanisms can be integrated into the process-level mechanisms for assessing the service qualities in its usage contexts defined via process models. This integration allows us to cover all the types of stakeholder involvement.

In future, we plan to implement all the mentioned models and mechanisms in detail and investigate the possibility to perform backwards transformation of the assessments of derived activities into the assessments of its base services. The whole set of assessment mechanisms is to be the foundation for higher-level policies aimed at solving particular problems of requirements engineering and architectural design such as requirements elicitation and verification, aligning the requirements to the capabilities of the organization (architecture and requirements negotiation), software architecture assessment and, in general, the support of using stakeholder assessments as driving forces for the software process.

**References:** 1. Amyot, D., Eberlein, A. An Evaluation of Scenario Notations and Construction Approaches for Telecommunication Systems Development // Telecommunication Systems. – 2003. – Vol. 24. – N 1. – P. 61-94. 2. Bosch, J. Design and Use of Software Architectures. – Reading: Addison-Wesley. – 2000. 3. Carroll, J.M. (ed.). Scenario-Based Design. – Wiley. – 1995. 4. Drew, P., Raymond, G., Weinberg, D. Talk and Interaction in Social Research Methods. – Sage Pubs. – 2006. 5. Dumas, M., Van der Aalst, W., ter Hofstede, A. (eds.). Process-Aware Information Systems. – Wiley-IEEE. – 2005. 6. Graham, I. Requirements Modelling and Specification for Service Oriented Architecture. – Wiley. – 2009. 7. Gregoriades, A., Sutcliffe, A. Scenario-Based Assessment of Nonfunctional Requirements // IEEE Trans. Software Eng. – 2005. – Vol. 31. – N 5. – P. 392-409. 8. Haumer, P., Heymans, P., Jarke, M., Pohl, K. Bridging the Gap Between Past and Future in RE: A Scenario-Based Approach // In Proc. RE'99. – IEEE CS Press. – 1999. – P. 66-73. 9. Kaschek, R., Kop, C., Shekhovtsov, V.A., Mayr, H.C. Towards Simulation-Based Quality Requirements Elicitation: A Position Paper // In REFSQ 2008. – LNCS, Vol. 5025. – Springer. – 2008. – P. 135-140. 10. Pastor, O., Molina, J.C. Model-Driven Architecture in Practice. – Springer. – 2007. 11. Rolland, C., Achour, C.B., Cauvet, C., et al. A proposal for a scenario classification framework // Requirements Engineering. – 1998. – Vol. 3. – N 1. – P. 23–47. 12. Seybold, C., Meier, S., Glinz, M. Scenario-Driven Modeling and Validation of Requirements Models // In Proc. SCESM'06. – ACM Press. – 2006. – P. 83-89. 13. Shekhovtsov, V., Kaschek, R., Zlatkin, S. Constructing POSE: a Tool for Eliciting Quality Requirements // In Proc. ISTA 2007. – LNI, Vol. P-107. – GI. – 2007. – P. 187–199. 14. Shekhovtsov, V.A. Interactive assessment of simulated service qualities by business stakeholders: principles and research issues // Проблеми програмування. – 2010, у друку.

**Н. А. БАЖЕНОВ**, аспирант НТУ «ХПИ»

## МОДЕЛИ УПРАВЛЕНИЯ КАЧЕСТВОМ СБОРА ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ НА ОСНОВЕ ТЕКСТОВ СПЕЦИФИКАЦИЙ

В статті пропонується підхід до керування якістю процесу розробки програмного забезпечення, побудований на підвищенні ефективності функціонування одного з етапів життєвого циклу програмного продукту – процесу збору вимог. Були виділені мети функціонування даного етапу, поставлені основні завдання, виявлені механізми взаємодії з іншими етапами життєвого циклу. Запропоновано концепцію керування якістю даного процесу на основі всебічного контролю якості вхідної й вихідної інформації, ресурсів і самого процесу.

В статье предлагается подход к управлению качеством процесса разработки программного обеспечения, основанный на повышении эффективности функционирования одного из этапов жизненного цикла программного продукта – процесса сбора требований. Были выделены цели функционирования данного этапа, поставлены основные задачи, выявлены механизмы взаимодействия с другими этапами жизненного цикла. Предложена концепция управления качеством данного процесса на основе всестороннего контроля качества входной и выходной информации, ресурсов и самого процесса.

In the article there has been proposed approach to quality management of software development process, based on the efficiency function increasing of one of the software product life cycle stages – requirements elicitation process. The purposes of functioning of the given stage have been established, the primary goals are put, the mechanism of interaction with other stages of life cycle is revealed. The concept of process quality management on the basis of total quality management of the input and output information, resources and the process is offered.

**Введение.** В настоящее время процесс разработки программного обеспечения представляет собой целый ряд организационных, технологических и технических групп задач [1, 2]. Эти группы задач формируют собой жизненный цикл (ЖЦ) программной системы (ПС), начиная от замысла создания такой системы, и заканчивая снятием с эксплуатации. Согласно [2], «*Модели жизненного цикла системы обычно разделяют на последовательные периоды реализации – стадии и этапы*». В данной работе предлагается рассмотреть один из таких этапов – процесс сбора требований к программному обеспечению.

В связи с постоянно возрастающей сложностью разрабатываемых программных систем, а также с повышающимися при этом ресурсоёмкостью и функциональными возможностями, на первый план выходит проблема обеспечения качества таких систем. Причём качество конечного продукта зависит не только от достижения определённых характеристик, заранее заданных заказчиком системы, но и от качества самого процесса разработки, состоящего из всех стадий жизненного цикла системы. Предлагается рассмотреть аспекты качества процесса разработки на примере стадии сбора

требований к программному обеспечению путём обработки спецификаций требований на естественном языке.

В разделе 1 описывается процесс сбора требований как одна из стадий ЖЦ ПС. Раздел 2 посвящен аспектам управления качеством стадии сбора требований. В разделе 3 рассмотрены основные выходные данные, являющиеся целью функционирования процесса сбора требований, а также возникающие при этом задачи. В 4-м разделе приводится схема функционирования процесса и основные функциональные блоки.

**1. Место процесса сбора требований в ЖЦ.** В соответствии с комплексом международных стандартов, регламентирующих жизненный цикл программных систем (ISO 12207 [3], ISO 15288 [4], ISO 9126-1 [5], ISO 25030 [6], SWEBOOK [7]), процесс сбора требований к программному обеспечению рассматривается как одна из дисциплин процесса разработки ПО (рис. 1).



Рис. 1. Процесс сбора требований в ЖЦ

Таким образом, объектом данного исследования является стадия технической группы процессов жизненного цикла на примере процесса сбора требований к программному обеспечению. Целью исследования ставится повышение эффективности управления качеством сбора требований к программному обеспечению путём разработки и исследования моделей и информационных технологий поддержки этапов жизненного цикла программных систем. Цель исследования основывается на комплексном использовании методов компьютерной лингвистики, искусственного интеллекта, теории автоматизированной обработки естественного языка, а также на теории многокритериальной оптимизации и принятия решений.

**2. Управление качеством процесса сбора требований.** Для управления качеством дисциплины жизненного цикла предлагается использовать процессный подход, или так называемый цикл Деминга Plan-Do-Check-Act [8]. Осуществляется управление измерениями качества – качеством входа, выхода, ресурсов, необходимых для данного этапа и самим процессом сбора требований (рис. 2).

Таким образом, получаем четыре задачи системной оптимизации для управления качеством входной информации, выходной информации, ресурсов и процесса сбора требований. Выделим критерии контроля качества по этим четырём задачам локальной оптимизации.

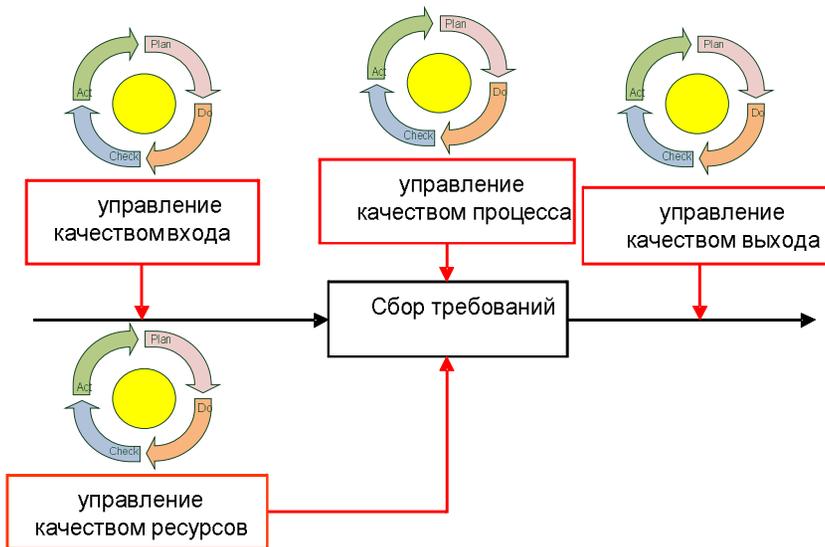


Рис. 2. Процесс сбора требований в ЖЦ

Контроль качества входной информации:

- Качество документов спецификаций требований. Включает в себя контроль качества по специализированным критериям, отсеивание нерелевантных документов, грубую фильтрацию (например, использование фильтров ключевых слов и фраз [9]).

Контроль качества выходной информации:

- Качество получаемой модели требований.
- Синхронизация управление качеством с этапом анализа и обработки требований.
- Поддержка возможности обратной связи.

Контроль качества ресурсов:

- Качество трудовых ресурсов.
- Качество финансовых ресурсов.
- Качество материальных ресурсов.
- Качество информационных ресурсов.

Качество процесса:

- Внутренние характеристики процесса.
- Управление различными вариантами реализации.

**3. Цели и задачи процесса сбора требований.** На вход дисциплины сбора требований поступают текстовые документы, выражающие спецификации требований к системе, нужды и пожелания заинтересованных

лиц (ЗЛ), заказчиков, потребителей. Процесс данной дисциплины должен проанализировать данную неструктурированную информацию и преобразовать её во множество требований ЗЛ, которое отражает взаимодействия между элементами системы, между системой и внешними интерфейсами, и эксплуатационные характеристики системы, удовлетворяющие нуждам ЗЛ.

Связь с дисциплиной анализа и обработки требований осуществляется по данным и обратной связи. В качестве этих данных выступает выходная информация дисциплины сбора требований, которая затем поступает на вход следующей дисциплины жизненного цикла системы. На преобразовании «вход-выход» осуществляется контроль качества информации по заранее заданным соответствующим метрикам. Обратное управляющее и корректирующее воздействие возможно как с последующей дисциплины, так и с остальных дисциплин жизненного цикла – проектирование, реализация. Согласно стандарту ISO 12207[3] выходная информации данной дисциплины должна содержать следующую информацию:

- Требуемые характеристики системы и условия использования сервисов.
- Системные ограничения.
- Соотношения между требованиями потребностями ЗЛ.
- Информация для определения системных требований.
- Информация для проверки конечных сервисов заданным характеристикам.
- Информация по поставке системы.

Так как объектом данного исследования является одна из технических стадий разработки программной системы, то организационные процессы жизненного цикла и соответствующие им аспекты не рассматриваются.

Таким образом, имеем взаимодействие дисциплины с другими этапами жизненного цикла на следующей схеме (рис. 3).

Для достижения необходимых результатов в виде перечисленной выходной информации перед дисциплиной ставятся следующие задачи:

- Идентификация ЗЛ. Определить отдельных ЗЛ или целые классы ЗЛ, вовлеченных в проект, на протяжении всего жизненного цикла.
- Идентификация требований:
  1. Извлечь требований ЗЛ, описывающие потребности, желания, ожидания и предполагаемые ограничения, выявленные ЗЛ.
  2. Определить системные ограничения, возникающие вследствие существующих соглашений, организационных и технических решений.
  3. Определить типовой набор действий для выявления всех необходимых функций, которые соответствуют ожидаемым сценариям и условиям работы системы.

4. Определить возможные взаимодействия между пользователями и системой.
  - Оценка требований. Проанализировать полное множество извлеченных требований.
  - Согласование требований:
    1. Решить проблемы и несогласованности в требованиях.
    2. Обеспечить обратную связь с ЗЛ для подтверждения того, что потребности и ожидания были адекватно извлечены и представлены.
    3. Предоставлять ЗЛ информацию о том, что их требования выражены корректно.
  - Обеспечить фиксацию и запись требований:
    1. Производить фиксацию требований ЗЛ в виде пригодном для управления требованиями на протяжении всего жизненного цикла системы и за его пределами.
    2. Поддерживать трассируемость требований ЗЛ и исходных источников ЗЛ.



Рис. 3. Взаимодействие процесса сбора требований с другими стадиями ЖЦ

**4. Функциональные элементы процесса сбора требований.** Исходя из поставленных задач перед процессом сбора требований, можно выделить основные функциональные элементы процесса, а также потоки данных и управляющих воздействий. На рис. 4 представлена общая схема функционирования процесса сбора требований.

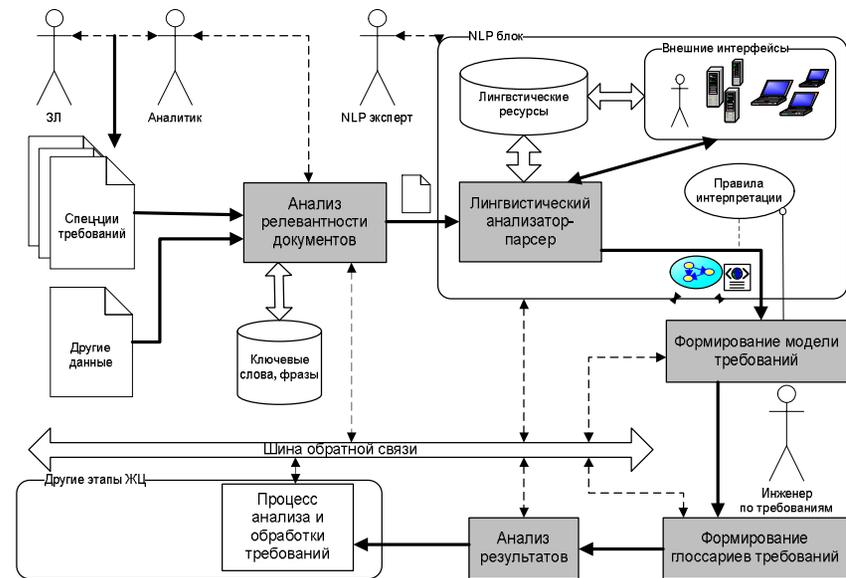


Рис. 4. Общая схема функционирования процесса сбора требований

На данной схеме показаны основные функциональные блоки серыми прямоугольниками, потоки данных сплошными стрелками, управляющие воздействия и обратные связи пунктирными стрелками, ресурсные взаимодействия фигурными стрелками.

На вход подсистемы сбора требований в блок анализа релевантности документов могут поступать следующие данные:

Спецификации требований, полученные в результате взаимодействия ЗЛ и аналитиков.

Другие документы, полученные от ЗЛ, или представляющие интерес для проектирования и создания программной системы.

Обратное взаимодействие от пары «ЗЛ–аналитик» о подтверждении/опровержении решения о релевантности документа/текста.

Управляющие и обратные воздействия от других элементов как процесса, так и ЖЦ в целом.

Анализ релевантности документов и их соответствие доменной специфичности разрабатываемой ПС определяется с учетом имеющихся ресурсов ключевых слов, фраз, и т.д. [9]. На выходе данного блока «фильтра грубой очистки» получается релевантный документ/текст, готовый к лингвистическому анализу.

Блок лингвистического анализа представляет собой конкретную реализацию одним из методов *Автоматизированной Обработки*

Естественного Языка (Natural Language Processing, NLP). Могут быть предложены следующие реализации:

- С помощью теории «Смысл-текст» [10] с использованием с использованием языкового онлайн-процессора «ЭТАП» [11].
- С помощью применения универсального структурированного языка Universal Network Language (UNL) [12] с использованием онлайн базы знаний UNL-KB [13].
- На основе теории контекстно-свободных грамматик с использованием статистического морфологического анализатора и основанного на правилах синтаксического парсера [14-16].
- На основе подхода «Ontological Semantics» [17].

При любой реализации внутри данного блока существует блок лингвистического анализа и парсинга, взаимодействующего с необходимыми ресурсами и внешними интерфейсами, конкретная реализация которых зависит от выбранного метода NLP. Связь с внешними интерфейсами может осуществляться путём взаимного обмена данными. Типичным набором задач лингвистического анализатора является [18]:

- Разбиение текста на отдельные лексемы (tokenization).
- Идентификация границ предложений.
- Морфологический анализ (POS tagging).
- Лемматизация (lemmatization).
- Присвоение всевозможных лингвистических категорий и свойств лексемам.
- Поверхностный синтаксический анализ и фразовая фрагментизация текста (chunking, shallow parsing).
- Идентификация вербальных категорий.
- Идентификация семантических ролей и связей.

Результатом работы данного блока является структурированный текст требований древовидного формата с корневой вершиной «текст», промежуточными узлами, характеризующими предложения и различные фразы, и с листьями, представленными отдельными лексемами с наборами присвоенных атрибутов.

Данный формат данных передаётся на блок формирования модели требований. Для сопоставления лингвистических элементов древовидного текста с конкретными концептами модели требований существует правила интерпретации (interpretation rules). Реализация конкретных правил зависит от выбора метода NLP, поэтому на схеме функционирования данный объект расположен в логической группе NLP блока. После формирования модели требований происходит заполнение глоссариев концептов требований по данной модели, в которой учитываются категории ЗЛ, множество их требований, выявленные системные ограничения, и другие данные.

Завершающим блоком процесса сбора требований является анализ полученных результатов. Полученные глоссарии проверяются на предмет неопределенности, неполноты, противоречивости и других несоответствий. Отметим, что данная проверка является первичной, так как более глубокий анализ осуществляется на последующих этапах ЖЦ.

Все функциональные блоки имеют управляющее взаимодействия с т.н. «шиной обратной связи», по средствам которой осуществляется возврат на предыдущие этапы с соответствующим управляющим воздействием, и корректировочные настройки функционирования этапов.

**Выводы.** Таким образом, рассмотренный процесс сбора требований к программному обеспечению является одним из важнейших этапов ЖЦ ПС. И для обеспечения качества ПС необходимо осуществлять управление качеством на различных элементах этого процесса путём оценивания отклонения от эталонных характеристик. Реализацию блока лингвистического анализа следует выбирать с учётом удовлетворения необходимых показателей качества функционирования всего процесса сбора требований.

**Список литературы:** 1. Андон Ф. И., Коваль Г. И., Коротун Т. М. и др. Основы инженерии качества программных систем.– К.: Академперіодика, 2007.– 680 с. 2. Луняев В. В. Процессы и стандарты жизненного цикла сложных программных средств.– М.: СИНТЕГ, 2006.– 276 с. 3. ISO/IEC Standard 12207:2008 System and software engineering – Software life cycle processes standard. 4. ISO/IEC 15288:2008 Systems and software engineering – System life cycle processes. 5. ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model. 6. ISO/IEC 25030:2007 Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality requirements. 7. ISO/IEC 19759:2005 Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOOK). 8. Репин В. В., Елиферов В. Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: РИА «Стандарты и качество», 2004. 9. Perkonigg M. Linguistische Aspekte des Attempto Controlled English (ACE). Masterarbeit.– Klagenfurt: Alpen-Adria-Universität Klagenfurt, 2009. 10. Мельчук И. Опыт теории лингвистических моделей Смысл-Текст.– М.: Наука, 1974. 11. Apresian J., Boguslavsky I., Iomdin L. etc. ETAP-3 Linguistic Processor.– Moscow: Laboratory of Computational Linguistics Institute for Information Transmission Problems, 2002. 12. J. Cardeñosa, A. Gelbukh, E. Tovar (Eds.). Universal Network Language: Advances in Theory and Applications – Research on Computing Science 12, 2005. 13. H. Uchida, M. Zhu. UNL2005 for Providing Knowledge Infrastructure.– Chiba, Japan: SeC2005 Workshop 2005. 14. Баженов М. О. Розробка алгоритмічного та програмного забезпечення обробки текстів на природній мові для підвищення ефективності застосування систем управління вимогами. Дипломна робота освітньо-кваліфікаційного рівня магістра.– Х.: НТУ «ХП», 2008.– 134 с. 15. Georg Weber. NIBA<tag> Aspekte der Implementierung eines erweiterten Taggers für die automatische Textannotation in NIBA: Masterarbeit.– Alpen-Adria-Universität Klagenfurt, 2007. 16. Günther Fliedl, Christian Kop, Jürgen Vöhringer. NIBA: Project & Toolset.– Alpen-Adria-Universität Klagenfurt, 2006. 17. Sergei Nirenburg, Victor Raskin.– The MIT Press, 2004. 18. Christopher D. Manning. Foundations of statistical natural language processing.– The MIT Press, 1999.

Поступила в редколлегию 15.12.09

**Б.Н.СОКОЛОВ**, аспирант НТУ «ХПИ»

### МОДЕЛИ УПРАВЛЕНИЯ КАЧЕСТВОМ ОБРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

У статті запропоновано підхід до управління якістю обробки вимог до програмного забезпечення, що базується на системному підході та стандартах ISO. Визначені цілі функціонування процесу аналізу та обробки вимог, розглянуті його характерні риси, особливості взаємодії з іншими процесами життєвого циклу програмних систем. Запропонована концепція управління якістю вхідною, вихідною інформацією та самим процесом аналізу та обробки вимог.

В статье предлагается подход к управлению качеством обработки требований к программному обеспечению, который основывается на системном подходе и стандартах ISO. Выделены цели функционирования процесса анализа и обработки требований, рассмотрены его основные свойства, особенности взаимодействия с другими процессами жизненного цикла программных систем. Предложена концепция управления качеством входной, выходной информацией и самим процессом анализа и обработки требований.

An approach to quality requirements management for software is outlined. This approach is based on system approach and ISO standards. Purposes of the system requirements analysis process activity were defined; its main properties and specific of interaction with other lifecycle processes were considered. The concept of quality control of the input, output information and the process of the system requirements analysis process was proposed.

**Введение.** Успех разрабатываемого программного обеспечения (ПО) в значительной степени зависит от того, насколько оно в конечном итоге удовлетворяет требования заинтересованных лиц (ЗЛ). Поэтому очень важно правильно собрать, формализовать и проанализировать требования ЗЛ по отношению к разрабатываемому продукту.

В [1] следующим образом определяются процесс программной инженерии и процесс жизненного цикла программной системы: «Под *процессом программной инженерии* понимается множество логически связанных видов деятельности по определению, проектированию и построению прикладных программных систем. Процесс программной инженерии имеет иерархическую архитектуру, включая множество *процессов жизненного цикла* программных систем. Эти процессы группируются по категориям – основные процессы, процессы поддержки, организационные процессы – и касаются аспектов не только собственно разработки программных систем, но и организации, контроля и управления разработкой».

Целью процесса анализа и обработки требований является преобразование требований, заданных заинтересованными лицами, во множество требуемых системных требований, которые будут использоваться в дисциплине проектирования [2].

Выявление системных требований к разрабатываемой системе происходит путем анализа информации об ее предполагаемом использовании. Спецификация системных требований должна включать в себя:

- Функции и возможности системы.
- Пользовательские, организационные и бизнес требования.
- Требования по безопасности, защищенности, эргономике, интерфейсу, производимым операциям, сопровождаемости.
- Архитектурные ограничения и требования по квалификации.

Ориентируясь на стандарты ISO [2,3] мы будем рассматривать разработку программного обеспечения как совокупность процессов. На рис. 1 изображены процессы, которые являются объектами исследования нашей группы.



Рис. 1. Процессный подход к управлению качеством ПО

Используя подход Деминга [4,5] получим схему управления качеством анализа обработки требований (рис. 2). Мы рассматриваем в данном случае 4 измерения качества, при этом считаем, что пятое измерение качества – качество программного продукта, по окончании полного цикла разработки, учитывать на данном этапе весьма сложно.



Рис. 2. Управление качеством анализа обработки требований

В данной статье мы рассмотрим взаимодействие процесса анализа и обработки требований с другими процессами жизненного цикла, обсудим его основные функциональные элементы и возможность оказания управляющих воздействий на них. В большей степени мы будем касаться управления качеством входа и управления качеством процесса.

**2. Взаимодействие с другими процессами жизненного цикла.** В соответствии со стандартом ISO/IEC 12207 [2] процесс анализа и обработки требований осуществляется после процесса определения требований и перед процессом проектирования. Процессы жизненного цикла связаны *прямой* и *обратной связью*. В случае, когда процедура контроля качества на входе одного процесса аналогична процедуре контроля качества на выходе другого процесса, принято использовать соответствующий блок контроля качества для контроля входных данных (чтобы избежать дублирования идентичных блоков).



Рис. 3. Взаимодействие процессов определения и анализа и обработки требований

На рис. 3 мы видим схему взаимодействия процессов определения требования и анализа и обработки требований. В соответствии с [1,2] в результате выполнения процесса определения требования мы получаем множество собранных требований ЗЛ и системных ограничений. При этом для получения этих данных могут быть применены различные подходы, основанные на теории распознавания текста, методах получения экспертных оценок, имитационном моделировании, психологии, социологии и т.д. При сборе требований возникает множество проблем, связанных, главным образом, с человеческим фактором. Ведь источником информации выступают заинтересованные лица, у каждого из которых могут быть свои личные интересы, которые обладают различным уровнем компетентности и т.д. Так в [6] выделяют следующие четыре типа возможных проблем при сборе требований ЗЛ: конфликты интересов ЗЛ; ограничения, связанные со способностями, компетентностью ЗЛ, а также нежелание ЗЛ сотрудничать;

неопределенность предпочтений ЗЛ; высокая сложность (например, большое количество требований, зависящих друг от друга).

На рис. 4 изображена схема взаимодействия процессов проектирования и анализа и обработки требований. В соответствии с ISO [2,3] в результате выполнения процесса анализа и обработки требований мы получаем структурированные требования заинтересованных лиц, результаты базового анализа этих требований и информацию по поддержке принятия решений по управлению проектом. Задачи принятия решений могут возникать во время каждого из этапов жизненного цикла [5,7-11]. В данном случае информация по поддержке принятия решений по управлению проектом включает в себя модели бизнес-процессов и решения по управлению проектом (основные этапы проекта, контрольные точки и т.д.). Модели бизнес-процессов могут быть использованы не только в качестве источника знаний по реализации определенного бизнес-процесса. Так в случае если разрабатываемая система имеет сервис-ориентированную архитектуру [12], модели бизнес-процессов играют важную роль для связывания различных сервисов между собой [13]. Для этих целей используются различные языки для формального описания бизнес-процессов и протоколов их взаимодействия между собой (такие как BPEL), а также системы управления бизнес-процессами [14].

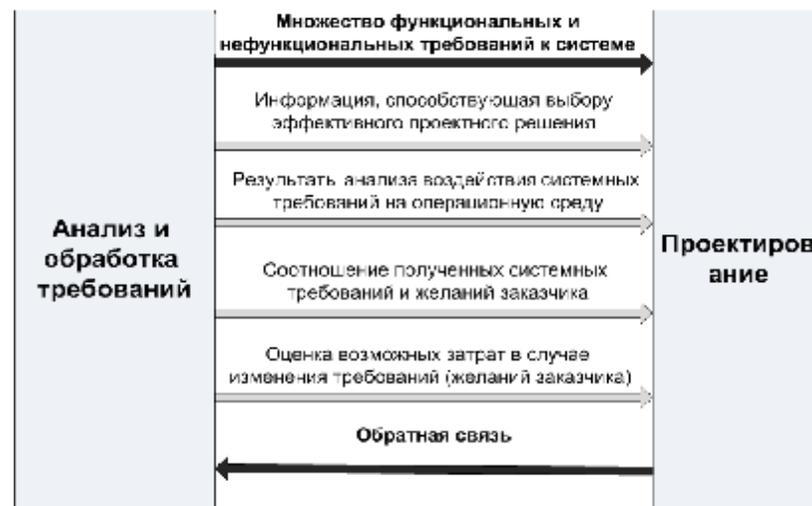


Рис. 4. Взаимодействие процессов анализа и обработки требований и проектирования

**3. Основные функциональные элементы анализа и обработки требований.** Изобразим основные функциональные элементы анализа и обработки требований на рис. 5. В общем случае возможна различная степень использования различных блоков (как блоков полностью, так и элементов из

которых они состоят). На выбор степени использования блоков будут влиять такие факторы, как размеры системы, ее тип (например, сервис-ориентированная), различные ресурсные ограничения. Для решения задач подобного рода может быть использован подход метод инжиниринга [15-17], который поможет на основании требований (основанных на ситуационных факторах компании) к процессу жизненного цикла и его элементам определить достаточную степень их использования.

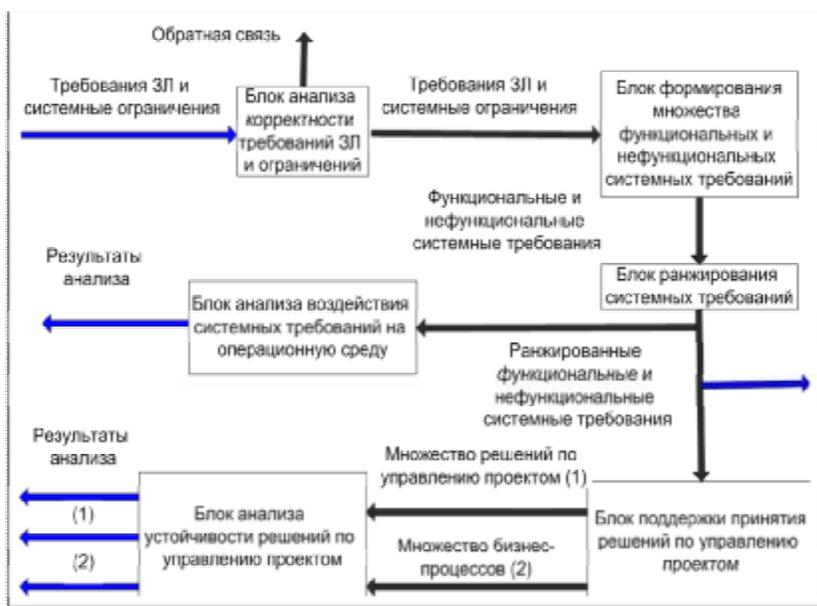


Рис. 5. Основные функциональные элементы анализа и обработки требований

Рассмотрим подробнее основные элементы процесса анализа и обработки требований. Как мы уже говорили ранее, в результате выполнения процесса определения требования мы получаем множество собранных требований ЗЛ и системных ограничений. Эти требования поступают на вход блока анализа корректности требований ЗЛ и ограничений (рис. 6). В этом блоке осуществляется проверка полноты и корректности требований, их тестируемости, осуществимости проектирования архитектуры системы, возможности использования и сопровождаемости системы. В общем случае проверка полноты требований зависит от формата, в котором они представлены, и зачастую сводится к синтаксическим проверкам [18]. Проверка корректности подразумевает под собой анализ связей между требованиями и поиск конфликтов. Анализ осуществимости проектирования

архитектуры системы призван показать, насколько сложно будет спроектировать систему на основании собранных требований, а также оценить вероятность появления ошибок на этапе проектирования. Для оценки операционных характеристик и характеристик сопровождаемости системы существуют специфические метрики (такие как соотношение решенных проблем за месяц к возникшим проблемам за месяц) [19, 20]. Однако, метрики эти, в большей степени, применимы в случаях, когда система уже разработана (либо к завершённым компонентам, прототипам системы и т.д.). В нашем случае доступны, главным образом, различные виды анализа рисков, связанных с использованием и сопровождением системы. Риски могут быть связаны с отклонением операционных и эргономических свойств системы от желаний заказчика; неточностью, нечеткостью и неполнотой системных требований, способными привести к частому внесению изменений в систему после ее выхода и т.д. Управление данным блоком осуществляется путем конфигурирования его настроек, связанных, по большей части, с изменением степени жесткости проверок. Очевидно, что чем жестче проверки, тем выше шанс, что требования будут возвращены на доработку на этап сбора требований, что приведет к дополнительным финансовым издержкам, однако, если условия будут слишком мягкими, повышаются риски возникновения ошибок при выполнении последующих процессов жизненного цикла.



Рис. 6. Блок анализа корректности требований заинтересованных лиц и ограничений

В блоке формирования множества функциональных и нефункциональных требований (рис. 7) осуществляется анализ и последующее преобразование требований ЗЛ и системных ограничений во множество функциональных и нефункциональных требований, поступающее на вход к процессу проектирования. Чем лучше будут произведены действия в этом блоке, тем легче будут проходить дальнейшие процессы жизненного цикла и тем меньше вероятность возникновения различных ошибок в финальном продукте. С другой стороны на функционирование блока накладываются ресурсные и временные ограничения. После того как были сформированы функциональные и нефункциональные системные требования, их, как правило, ранжируют. Довольно часто разработчик программного продукта не имеет возможности в полной мере удовлетворить требования всех ЗЛ в рамках существующих ресурсных и временных ограничений. В результате

получается задача определения подмножества требований ЗЛ, реализация которых с одной стороны вписывалась в ресурсные рамки, а с другой стороны позволяла бы создать продукт, удовлетворяющий ЗЛ. Для ранжирования требований могут использоваться как метрики, показывающие соотношение относительной важности требования к его относительной стоимости [21], так и более сложные показатели, учитывающие время на разработку отдельных модулей и т.д. [22].

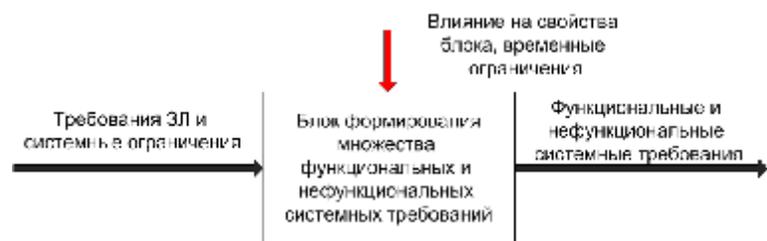


Рис. 7. Блок формирования множества функциональных и нефункциональных системных требований

Блок поддержки принятия решений по управлению проектом (рис. 8) предоставляет возможность на основании требований к системе получать модели бизнес-процессов и множество решений по управлению проектом. Для данного блока необходимы программные средства для повторного использования бизнес-процессов и решений по управлению проектом, включающие в себя соответствующие репозитории, конверторы между различными форматами и подсистемы поиска.

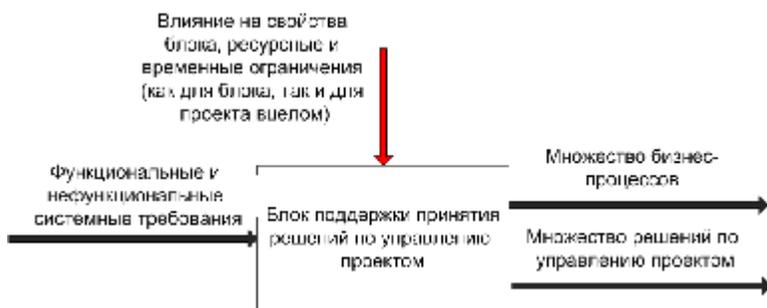


Рис. 8. Блок поддержки принятия решений по управлению проектом

**Выводы.** Мы рассмотрели процесс анализа и обработки требований к ПО, качество выполнения которого, так же как и других процессов жизненного цикла, в определенной степени влияет на качество получаемого программного продукта. Управление качеством данного процесса

осуществляется путем варьирования степени использования его блоков (и их элементов), наложением различных ресурсных и временных ограничений на выполнение определенных операций в блоках (и их элементах), а также заданием различных настроек для блоков (таких как уровень жесткости проверки качества входных данных).

В дальнейшем планируется разработать процедуры анализа корректности требований ЗЛ и системных ограничений, процедуру трансформации требований ЗЛ в функциональные и нефункциональные системные требования, прототип ПО для повторного использования бизнес-процессов, прототипа ПО для повторного использования решений по управлению проектом.

**Список литературы:** 1. Ф.И. Андон, Г.И. Коваль, Т.М. Коротун, Е.М. Лаврищева, В.Ю. Суслов. Основы инженерии качества программных систем 2 изд. перераб. и доп. – Киев: Академперіодика, 2007. 2. ISO/IEC 12207:2008 System and software engineering – Software life cycle processes. 3. ISO/IEC 15288:2008 Systems and software engineering — System life cycle processes. 4. W. Edwards Deming. Out of the Crisis. – Cambridge: MIT Press, Center for Advanced Engineering Study, 1986. 5. V. Basili, G. Caldiera. Improve Software Quality by Reusing Knowledge and Experience, Sloan Management Review, MIT Press, vol. 37(1), 1995 – p. 55-64. 6. S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, P. Grünbacher. Value-Based Software Engineering. – Berlin: Springer-Verlag Berlin Heidelberg, 2006. 7. R.Kazman, M.Klein, P.Clemens. ATAM: Method of Architecture Evaluation. Technical Report// CMU/SEI-2000-TR-004, CMU, 2000. 8. M.Svahnberg, C.Wholin, L.Lundber. A Quality-Driven Decision-Support Method for Identifying Software// Int. Journal of Software Engineering and Knowledge Engineering, vol. 13(5), 2003 – p. 547-573. 9. R. Ocker, S. R. Hiltz, M. Turoff, J. Fjermestad. The effects of distributed group support and process structuring on software requirements development teams: results on creativity and quality// Journal of Management Information Systems, vol. 12(3), December 1995 – p. 127-153. 10. L. Liu, E. Yu, J. Mylopoulos. Security and privacy requirements analysis within a social setting// Requirements Engineering Conference, 2003. Proceedings. 11th IEEE International – p. 151-161. 11. N.A.M. Maiden, G. Rugg. ACRE: selecting methods for requirements acquisition// Software Engineering Journal, vol. 11(3), May 1996 – p. 183-192. 12. Th. Erl. Service-Oriented Architecture: Concepts, Technology, and Design. – NJ: Pearson Education, 2005. 13. M. Wheske. Business Process Management. Concepts, Languages, Architectures. – Berlin: Springer-Verlag Berlin Heidelberg, 2007. 14. Д. Грязнов. Система управления бизнес-процессами как элемент процессного подхода к управлению бизнесом// Intelligent Enterprise, №4, 2008 – с. 12-15. 15. A.F. Harmsen. Situational Method Engineering// Doctoral dissertation. – Twente: University of Twente, 1997. 16. S. Brinkemper, I. van de Weerd, M. Saeki, J. Versendaal. Process Improvement in Requirements Management: A Method Engineering Approach// Requirements Engineering Foundation for Software Quality. – Berlin: Springer-Verlag Berlin Heidelberg, 2008 – p. 6-22. 17. J. Ralyté, C. Rolland. An Assembly Process Model for Method Engineering// Advanced Information Systems Engineering. – Berlin: Springer-Verlag Berlin Heidelberg, 2001 – p. 267-283. 18. J. E. Burge, J. M. Carroll, R. McCall, I. Mistrik. Rationale-Based Software Engineering. – Berlin: Springer-Verlag Berlin Heidelberg, 2008. 19. St. H. Kan. Metrics and Models in Software Quality Engineering, Second Edition. – Boston: Pearson Education, 2003. 20. Kh. El Emam. The ROI from software quality. – Boca Raton, FL: Auerbach Publications Taylor & Francis Group, 2005. 21. J. Karlsson, K. Ryan. A cost-value approach for prioritizing requirements// IEEE Software, vol. 14(5), 1997 – p. 67-74. 22. Bj. Regnell, M. Höst, J. Natt, P. Beremark, Th. Hjelm. An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software// Requirements Engineering, Springer-Verlag London Limited, vol. 6 (1), 2001 – p. 51-62.

Поступила в редакцию 15.12.09

*М. М. МАЛЯР*, канд. техн. наук

### ВИЗНАЧЕННЯ ТА ЦІЛЕСПРЯМОВАНА ЗМІНА ВАГОВИХ КОЕФІЦІЄНТІВ ПРИ БАГАТОКРИТЕРІАЛЬНОМУ ВИБОРІ

В статті описані методи визначення вагових коефіцієнтів і пропонується процедура, що дозволяє в автоматизованому режимі змінювати їх значення.

В статье описаны методы определения весовых коэффициентов и предлагается процедура, которая позволяет в автоматизированном режиме изменять их значения.

In the paper methods of the weight factors determination are described. And procedure of their significance change in automatic regimen is proposed.

**Вступ.** Результат діяльності в реальному житті майже неможливо оцінити одним числом. Наприклад, покупка машини. Покупець хоче, щоб ціна була невисока, машина надійна, дизайн хороший, експлуатація невисока і т.д. Всі ці характеристики між собою зв'язані і зміна однієї веде до зміни інших. Яким чином розв'язати таку проблему. Розв'язання таких проблем можлива за допомогою багатокритеріальних оптимізаційних моделей.

Багатокритеріальні моделі широко використовуються у всіх сферах людської діяльності. Опишемо дану модель наступним чином: нехай задана множина альтернатив  $X$ , кожна альтернатива цієї множини оцінюється за допомогою декількох показників ефективності. Необхідно вибрати найкращу альтернативу, або проранжувати їх у порядку привабливості.

Множина альтернатив  $X$  може задаватись континуальною множиною, тобто у вигляді області обмежень, або дискретно. Критерії (показники) ефективності можуть бути описані за допомогою цільових функцій або задані числовими оцінками. В подальшому будемо позначати їх  $f_1(x), f_2(x), \dots, f_k(x)$ , де  $k$  – кількість показників.

Кінцевою ціллю задачі вибору являється відшукування найкращого або "оптимального" розв'язку. Тому під оптимізацією розуміють відшукування альтернативи, яка переважає всі інші альтернативи в певному розумінні. Математична форма запису даної задачі наступна: знайти  $\text{extr}_{x \in X} \{f_1(x), f_2(x), \dots, f_k(x)\}$ .

Модель, що описує множину допустимих розв'язків є об'єктивною, але оцінка якості розв'язку проводиться за багатьма критеріями. Щоб вибрати найкращий результат потрібно знайти компроміс між оцінками за різними критеріями, тобто необхідна додаткова інформація. Такою інформацією служать ваги критеріїв, які показують важливість критеріїв і повинні або можуть залежати від значень самих критеріїв.

**Постановка задачі.** Найбільш поширеним методом розв'язування багатокритеріальних задач є метод згортки векторного критерію. Суть якого полягає у побудові скалярної функції  $F$ , яка являється узагальненим критерієм відносно векторного критерію та вектора ваг і багатокритеріальний вибір зводиться до розв'язання однокритеріальної задачі оптимізації  $\text{extr}_{x \in X} F(\alpha, f(x))$ , де  $f(x) = \{f_1(x), f_2(x), \dots, f_k(x)\}$ ,  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$  – вагові

коефіцієнти відносної важливості критеріїв  $f_1(x), f_2(x), \dots, f_k(x)$ .

В якості узагальнених критеріїв, наприклад, можуть бути використані відомі згортки наступного виду:

1. Адитивна  $F(\alpha, f(x)) = \sum_{i=1}^k \alpha_i f_i(x)$ .
2. Мультиплікативна  $F(\alpha, f(x)) = \prod_{i=1}^k f_i^{\alpha_i}(x)$ .
3. Середньо степенева  $F(\alpha, f(x)) = \left[ \frac{1}{k} \sum_{i=1}^k \alpha_i f_i(x) \right]^{1/k}$ .

Таким чином, вибравши узагальнений критерій оптимальності, ми від суб'єктивізму при наданні переваги між векторними критеріями переходимо до суб'єктивізму, який пов'язаний з призначенням числових значень ваговим коефіцієнтам  $\alpha_i (i = \overline{1, k})$ . Як відзначається у роботі [1], визначення вагових коефіцієнтів являється двох етапною проблемою, тобто попереднє визначення і зміна цих значень. Тому виникають питання: "Яким чином задати числові значення ваговим коефіцієнтам?" і "Яким чином їх відкоригувати у разі необхідності?"

Відповідь на ці запитання в деякій степені можна отримати, якщо є додаткова інформація про важливість кожного критерію ефективності.

В подальшому будемо вважати, що вагові коефіцієнти є нормовані, тобто виконується додаткова умова  $\sum_{i=1}^k \alpha_i = 1$  і величина  $\alpha_i$  визначає важливість  $i$ -

го критерію ефективності над іншими критеріями і задається кількісною оцінкою додатного значення.

**Визначення вагових коефіцієнтів.** Аналіз існуючих способів визначення вагових коефіцієнтів показує, що найбільш розповсюдженими є підходи, які використовують ранжування або попарне порівняння.

Приведемо деякі відомі способи визначення ваг [1, 2]. Для рівноцінних критеріїв значення ваг вибираються однаковими  $\alpha_i = 1/k, i = \overline{1, k}$ .

Для нерівноцінних критеріїв ваги вибираються у відповідності з важливістю критерію при виконанні умов:

$$\alpha_i > 0, \sum_{i=1}^k \alpha_i = 1, i = \overline{1, k}. \quad (1)$$

Спосіб 1. Для кожного критерію оптимальності обчислюється коефіцієнт  $\alpha_i$ , який визначає максимально можливе відхилення по  $i$ -му критерію

$$\alpha_i = \frac{f_i^M - f_i^m}{f_i^M} = 1 - \frac{f_i^m}{f_i^M}, \quad (2)$$

де  $f_i^m = \min_{x \in X} f_i(x)$ ,  $f_i^M = \max_{x \in X} f_i(x)$ .

Вагові коефіцієнти критеріїв обчислюються як  $w_i = \alpha_i (\sum_{i=1}^k \alpha_i)^{-1}$ ,  $i = \overline{1, k}$ , і отримують найбільше значення для тих критеріїв, відносне відхилення яких найбільш значуще.

Спосіб 2. Нехай всі  $f_i^m \neq 0$  ( $\overline{1, k}$ ), тоді обчислюємо величини

$$\beta_i(x) = \frac{f_i(x) - f_i^m}{f_i^m}, \quad (3)$$

які характеризують відхилення критерію оптимальності від його найменшого значення.

Припустимо, що важливість  $i$ -го критерію оптимальності залежить від виконання нерівності

$$\beta_i(x) \leq \xi_i. \quad (4)$$

Величина  $\xi_i$  задається з умови, що чим важливіший критерій, тим менше вибирається значення  $\xi_i$ .

Розв'яжемо для кожного критерію оптимальності наступну задачу:

$$y^i = \arg \min_{x \in X} f_i(x), \quad i = \overline{1, k}. \quad (5)$$

Нехай  $R_i^*$  – найбільший радіус кулі, яка побудована навколо точки мінімуму  $i$ -го критерію, всередині якої точки  $X$  задовольняють умові (4):

$$R_i^* = \max_{x \in X} \left\{ \sum_{j=1}^n (x_j - y_j^i)^2 \right\}, \quad (6)$$

при умові  $\beta_i(x) \leq \xi_i$ , де  $n$  – розмір простору змінних  $X$ .

Чим більший радіус кулі  $R_i^*$ , в якому відносне відхилення  $i$ -го критерію від його мінімального значення не переважає  $\xi_i$ , тим менше потрібно вибирати значення вагового коефіцієнта.

Спосіб 3. Нехай маємо  $k$  критеріїв ефективності, що описуються цільовими функціями  $f_1(x), f_2(x), \dots, f_k(x)$ .

Побудуємо матрицю  $A$  елементи якої обчислюються наступним чином

$$a_{ij} = \left| \frac{f_j^m - f_j(y_i)}{f_j^m} \right|. \quad (7)$$

Ці величини визначають відносне відхилення оптимального значення  $j$ -го критерію  $f_j^m$  від його значення, яке отримано при оптимальному розв'язку  $y_i$  для  $j$ -го критерію оптимальності  $f_j(y_i)$ . Очевидним є факт, що  $a_{ii} = 0$ ,  $a_{ij} \geq 0$ .

Для кожного стовпця матриці  $A$  знаходимо максимальне і мінімальне (ненульове) значення  $b_j = \max_i \{a_{ij}\}$ ,  $c_j = \min_i \{a_{ij}\}$ ,  $i = \overline{1, k}$ .

Після чого обчислюємо величини  $d_j = b_j - c_j$ , а вагові коефіцієнти вибираються пропорційно значенням  $d_j$

$$\alpha_j = [d_j] \left( \sum_{j=1}^k [d_j] \right)^{-1}, \quad j = \overline{1, k}, \quad (8)$$

де знак  $[e]$  означає ціле значення числа  $e$ .

Якщо, наприклад, у  $l$ -му стовпчику значення  $f_l(y_i)$  сильно відрізняються від  $f_l^*$ , тобто значення  $a_{il}$  мають великі значення, то ваговий коефіцієнт повинен, бути взятий великий.

Спосіб 4. Т. Сааті [3] запропонував мультиплікативну модель обчислення ваг критеріїв. Ідея даного підходу полягає в тому, що вагові коефіцієнти визначаються за допомогою процедури, яка аналогічна зважуванню “на око”, тобто дається оцінка експертів у скільки разів один критерій важливіше іншого. Для цього він запропонував наступні оцінки:

- $p_{ij} = 1$ , якщо  $f_i$  еквівалентний  $f_j$ ;
- $p_{ij} = 3$ , якщо  $f_i$  переважає  $f_j$ ;
- $p_{ij} = 5$ , якщо  $f_i$  суттєво переважає  $f_j$ ;
- $p_{ij} = 7$ , якщо  $f_i$  абсолютно переважає  $f_j$ ;
- $p_{ij} = 9$ , якщо  $f_i$  безумовно переважає  $f_j$ .

Значення  $p_{ij} = 2, 4, 6, 8$  вважаються проміжними основних градаций. Діагональні елементи матриці парних порівнянь  $p_{ii} = 1$ , а  $p_{ji} = 1/p_{ij}$ .

Ваги порівнюваних об'єктів обчислюються наступним чином:

$$\lambda_i = k \sqrt[k]{\prod_{j=1}^k p_{ij}}, \quad \alpha_i = \frac{\lambda_i}{\sum_{i=1}^k \lambda_i}, \quad i = \overline{1, k}. \quad (9)$$

**Спосіб 5.** Дуже часто інформація про важливість критеріїв оптимальності може бути задане за допомогою матриці парних порівнянь  $A$ .

Для побудови матриці парних порівнянь можуть бути задані різні типи калібровок переваг. Опишемо деякі з них:

1. Проста структура:

$$\forall i, j; i \neq j, a_{ij} = \begin{cases} 1, & \text{якщо } f_i \text{ переважає } f_j, \\ 0, & \text{якщо } f_j \text{ переважає } f_i, \\ 1/2, & \text{якщо } f_i \text{ еквівалентний } f_j. \end{cases}$$

2. Турнірна:  $\forall i, j; a_{ij} \geq 0; a_{ij} + a_{ji} = \text{const}$ .

3. Степенева:  $\forall i, j; a_{ij} > 0; a_{ij} * a_{ji} = 1$ .

4. Косометрична:  $\forall i, j; a_{ij} + a_{ji} = 0$ .

5. Імовірнісна:  $\forall i, j; 0 \leq a_{ij} \leq 1; a_{ij} + a_{ji} = 1$ .

Існують схеми переходів між типами калібровок.

Після побудови матриці парних порівнянь вагові коефіцієнти можуть бути обчислені наступним чином:

$$\alpha_i = \sum_{j=1}^k a_{ij} \left( \sum_{i=1}^k \sum_{j=1}^k a_{ij} \right)^{-1}. \quad (10)$$

**Алгоритм модифікації вагових коефіцієнтів.** На сьогоднішній день відомо багато підходів, методів та алгоритмів щодо модифікації значень вагових коефіцієнтів. Автором пропонується підхід інтерактивного корегування ваг [4], суть якого полягає у тому, що він надає можливість одночасно змінювати і залишати незмінними декілька вагових коефіцієнтів.

Загальна схема підходу виглядає наступним чином:

1. Нехай значення вагових коефіцієнтів визначені: задані або обчислені.

2. У діалоговому режимі будуватиметься множина величин  $W = \{W_i\}, i = \overline{1, k}$ :

$$W_i = \begin{cases} 1, & \text{якщо вагу } i\text{-го критерія потрібно замінити,} \\ 0, & \text{якщо вагу } i\text{-го критерія потрібно залишити,} \\ -1, & \text{в іншому випадку.} \end{cases}$$

3. На основі множини  $W$  будуються множини величин  $U = \{U_i\}, V = \{V_i\}, (i = \overline{1, k})$

$$U_i = \begin{cases} 1, & \text{якщо } W_i = 1, \\ 0, & \text{в іншому випадку.} \end{cases} \quad V_i = \begin{cases} 1, & \text{якщо } W_i = 0, \\ 0, & \text{в іншому випадку.} \end{cases}$$

4. Визначаються нові значення для ваг критеріїв за наступною формулою:

$$\alpha_i^H = \frac{(1 - \sum_{j=1}^N \alpha_j^C V_j + \varepsilon U_i) \alpha_i^C}{1 - \sum_{j=1}^N \alpha_j^C V_j - \frac{\varepsilon (V_i - 1) \sum_{j=1}^N \alpha_j^C U_j}{1 - \sum_{j=1}^N \alpha_j^C V_j}}, \quad (11)$$

де  $\alpha_i^H, \alpha_i^C$  – відповідно нові і старі значення ваг. Величина  $\varepsilon$  задається користувачем і від неї залежить зміна ваг. Якщо  $\varepsilon > 0$ , то відповідні ваги збільшуються, а якщо  $\varepsilon < 0$  – то зменшуються.

**Висновки.** Як показують дослідження проведені по взаємозв'язку людини з комп'ютером, людина здатна вказати значення якого критерію вона хотіла би збільшити, а якого зменшити. І це можливо через зміну ваг. Ця задача є складною, оскільки наслідки зміни ваг непередбачувані. Наприклад, людина не може знати на скільки змінити ваги критеріїв, щоб результат став задовільним.

**Список літератури:** 1. Айвазян С.А., Бухштабер В.М. и др. Прикладная статистика: классификация и снижение размерности – М.: Финансы и статистика, 1989. – 607 с. 2. Лотов А.В., Поспелова И.И. Многокритериальные задачи принятия решений: Учебное пособие. – М.: МАКСПресс, 2008. – 197 с. 3. Саати Т. Принятие решений. Метод анализа иерархий. – М.: Радио и связь, 1993. – 278 с. 4. Маляр М.М. Деякі алгоритми знаходження ефективного розв'язку у багатокритеріальних задачах лінійного програмування. – “Теорія обчислень”. Зб. наук. праць НАН України ІК ім. В.М. Глушкова. – Київ – 1999. – С. 252-254.

Надійшла до редколегії 15.12.09

**Л. С. ЛОБАНОВА**, канд. фіз.-мат. наук, доц. каф. прикладної математики УПА (м. Харків),  
**В. О. ПАСІЧНИК**, канд. техн. наук,  
**О. О. ЧЕРНЯК**, аспірант УПА

### ОПТИМІЗАЦІЯ ОБЧИСЛЕНЬ В ІНТЕГРАЛЬНОМУ МЕТОДІ НАЙМЕНШИХ КВАДРАТІВ НАБЛИЖЕННЯ ФУНКЦІЙ ОДНІЄЇ ТА ДВОХ ЗМІННИХ

Пропонується метод, що дозволяє оптимізувати обчислення в інтегральному методі найменших квадратів і базується на апроксимації функцій однієї та двох змінних В-сплайнами першого степеня. Отримані явні формули для подвійних інтегралів від добутку функцій, які є сумами В-сплайнів з різними носіями.

Предлагается метод, позволяющий оптимизировать вычисления в интегральном методе наименьших квадратов и базирующийся на аппроксимации функций одной или двух переменных В-сплайнами первой степени. Метод позволяет получить явные формулы для двойных интегралов от произведения функций, которые являются суммами В-сплайнов с отличающимися носителями.

Is offered the method, allowing to optimise calculation in an integrated method of the least squares and based on approximation of functions of one or two variables by B-splines of the first degree. The method allows to receive obvious formulas for double integrals from product of functions which are the sums B-splines with differing carriers.

**Вступ.** Проекційно-сітковий метод (метод скінченних елементів) став в наш час ефективним методом розв'язання різноманітних задач математичної фізики. Ця обставина в значній мірі обумовлена досягненнями математики в області теорії проекційних методів та теорії апроксимації за допомогою функцій із скінченними носіями (фінітних функцій), а також швидким ростом застосування електронно-обчислювальної техніки в наукових дослідженнях. На протязі багатьох років варіаційні методи, що є частинним випадком проекційних, використовуються для розв'язання задач математичної фізики. Суть цих методів полягає у формулюванні задачі в варіаційній формі як задачі про пошук функції, яка реалізує мінімум, або, в загальному випадку, екстремум деякого функціонала, і в наступному знаходженні наближення до цієї функції. При цьому зручними виявляються такі алгоритми наближеного розв'язування задач, які, з одного боку, по формі були б варіаційними, або проекційними і, таким чином, мали всі їх переваги, а з іншого боку, щоб ці алгоритми приводили до систем рівнянь, подібних до тих, які виникають в різницевих методах (незначна кількість елементів матриць цих систем були б відмінні від нуля). Такими алгоритмами є проекційно-сіткові алгоритми (метод скінченних елементів). Щоб прийти до таких алгоритмів достатньо в

варіаційних або проекційних методах в якості базисних функцій  $\{j_i\}$  брати функції із скінченними носіями, тобто такі функції, які відмінні від нуля лише на невеликій частині тієї області, на якій визначений шуканий розв'язок задачі. Привабливими рисами проекційно-сіткових методів є наступні: шукані коефіцієнти часто мають ясну змістовну інтерпретацію, наприклад, є значеннями шуканого розв'язку у вузлах сітки; за допомогою фінітних базисних функцій вдається інколи легко врахувати геометрію області і тим самим усунути труднощі, які виникають в різницевому методі при задовільненні граничним умовам. Крім того, якщо вдало вибран проекційний алгоритм і базисні функції, то подальший процес побудови розв'язку відбувається «автоматично» і чисельно реалізується, наприклад, за допомогою існуючих систем комп'ютерної математики (Mathcad, Matlab, Maple).

**Аналіз літератури.** Із загальної теорії наближених методів Л. В. Канторовича [1] витікає загальна схема побудови наближених методів, в яку в багатьох випадках вкладаються методи Рітца, Бубнова-Гальоркіна і деякі їх узагальнення, метод найменших квадратів, інтегральний метод найменших квадратів. Важливі аспекти чисельної реалізації методу найменших квадратів висвітлені в роботі [2]. Загальні умови збіжності методу найменших квадратів і ряд його застосувань наведені в монографії [3].

**Мета статті.** Метод найменших квадратів у інтегральній формі полягає у наближенні функції  $z = f(x, y), (x, y) \in D$  сумою  $S_N(x, y) = \sum_{k=1}^N C_k j_k(x, y)$  шляхом розв'язання мінімізаційної задачі  $J(C_1, \mathbf{L}, C_N) \rightarrow \min_{C_1, \mathbf{L}, C_N}$ , де

$$J(C_1, \mathbf{L}, C_N) = \iint_D \left[ f(x, y) - \sum_{k=1}^N C_k j_k(x, y) \right]^2 dx dy.$$

Тут  $j_1, \mathbf{L}, j_N$  – деяка повна система функцій. Зокрема, якщо ця система функцій ортонормована  $(j_i, j_k) = d_{i,k}, 1 \leq k, i \leq \infty, d_{i,k} = \begin{cases} 1, i = k \\ 0, i \neq k \end{cases}$ , то

$$C_k = \iint_D f(x, y) j_k(x, y) dx dy, k = \overline{1, N}. \quad (1)$$

Якщо функції  $j_k(x, y)$  є швидкоосцилюючими функціями (наприклад  $j_k(x, y) = \cos 2p(k_1 x + k_2 y)$ ), то обчислення інтегралів (1) за допомогою класичних квадратурних формул (прямокутників, парабол тощо) не є ефективним. У цьому випадку ефективними є формули, побудовані шляхом

заміни функції  $f(x, y)$  відповідним сплайном і подальшим точним обчисленням отриманих інтегралів.

Ця ідея може бути плідною навіть для випадку, коли  $j_k(x, y)$  не мають сильної осциляції, а є сплайнами. У цьому випадку, якщо  $j_k(x, y)$  є сплайнами степеня  $r_1$  і  $f(x, y)$  замінена також сплайном степеня  $r_2$ , то для коефіцієнтів  $C_k, k = \overline{1, N}$  можна отримати систему, елементи матриці якої можна знайти точно. Метою даної статті є дослідження обчислювальних аспектів обчислення вказаних елементів матриці.

**Основна частина.** Нехай функція двох змінних  $f(x, y)$  задана своїми значеннями  $z_{i,j} (i = \overline{1, M}; j = \overline{1, N})$  на сітці  $\Delta_1 : a = x_1 < x_2 < \dots < x_M = b; c = y_1 < y_2 < \dots < y_N = d$ . Застосуємо для її апроксимації систему неперервних кусково-лінійних функцій  $\{h(x, X_i)\}$ ,  $X_i = \{x_{i-1}, x_i, x_{i+1}\}$  з наступними властивостями:  $h(x_i, X_i) = 1$ ,  $h(x_j, X_i) = 0 (j \neq i)$ ;  $h(x, X_i)$  відмінна від нуля лише на проміжку  $(x_{i-1}, x_{i+1})$  і визначається рівностями

$$h(x, X_i) = \begin{cases} 0, & -\infty < x \leq x_{i-1}; x_{i+1} \leq x < \infty \\ \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} < x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x_i < x < x_{i+1} \end{cases};$$

і аналогічну систему  $\{h(y, Y_j)\}$  по змінній  $y$ , де  $Y_j = \{y_{j-1}, y_j, y_{j+1}\}$ .

Тоді дану функцію  $f(x, y)$  можна наблизити білінійним сплайном

$$Sp_{M,N}(x, y) = \sum_{k=1}^M \sum_{l=1}^N z_{k,l} h(x, X_k) h(y, Y_l). \quad (2)$$

Розглянемо тепер сітку з іншим кроком  $\Delta_2 : (x_i^*, y_j^*) (i = \overline{1, m}, j = \overline{1, n})$  і побудуємо на ній теж білінійний сплайн

$$Sp_{m,n}(x, y) = \sum_{i=1}^m \sum_{j=1}^n C_{i,j} h(x, X_i^*) h(y, Y_j^*); X_i^* = \{x_{i-1}^*, x_i^*, x_{i+1}^*\}, Y_j^* = \{y_{j-1}^*, y_j^*, y_{j+1}^*\}.$$

Знайдемо коефіцієнти розкладу  $C_{i,j} (i = \overline{1, m}, j = \overline{1, n})$  з умови:

$$J(z, C) = \int_a^b \int_c^d (Sp_{M,N}(x, y) - Sp_{m,n}(x, y))^2 dx dy \rightarrow \min_{C_{i,j}}.$$

Користуючись необхідною умовою екстремуму  $\frac{\partial J(z, C)}{\partial C_{r,s}} = 0, (r = \overline{1, m}; s = \overline{1, n})$ , отримуємо систему лінійних алгебраїчних

рівнянь відносно  $C_{i,j} (i = \overline{1, m}; j = \overline{1, n})$ . Дійсно,

$$\frac{\partial J(z, C)}{\partial C_{r,s}} = -2 \int_a^b \int_c^d (Sp_{M,N}(x, y) - Sp_{m,n}(x, y)) h(x, X_r^*) h(y, Y_s^*) dx dy = 0,$$

$$\int_a^b \int_c^d \left( Sp_{M,N}(x, y) - \sum_{i=1}^m \sum_{j=1}^n C_{i,j} h(x, X_i^*) h(y, Y_j^*) \right) h(x, X_r^*) h(y, Y_s^*) dx dy = 0.$$

Звідси

$$\sum_{i=1}^m \sum_{j=1}^n C_{i,j} \int_a^b \int_c^d h(x, X_i^*) h(y, Y_j^*) h(x, X_r^*) h(y, Y_s^*) dx dy = \int_a^b \int_c^d Sp_{M,N}(x, y) h(x, X_r^*) h(y, Y_s^*) dx dy$$

Тобто, вказана система лінійних алгебраїчних рівнянь має вигляд

$$\sum_{i=1}^m \sum_{j=1}^n a_{i,j,r,s} C_{i,j} = b_{r,s} (r = \overline{1, m}; s = \overline{1, n}), \quad (3)$$

де

$$a_{i,j,r,s} = \int_a^b \int_c^d h(x, X_i^*) h(x, X_r^*) h(y, Y_j^*) h(y, Y_s^*) dx dy = \quad (4)$$

$$= \int_a^b h(x, X_i^*) h(x, X_r^*) dx \int_c^d h(y, Y_j^*) h(y, Y_s^*) dy;$$

$$b_{r,s} = \int_a^b \int_c^d Sp_{M,N}(x, y) h(x, X_r^*) h(y, Y_s^*) dx dy = \quad (5)$$

$$= \sum_{k=1}^M \sum_{l=1}^N z_{k,l} \int_a^b \int_c^d h(x, X_k) h(x, X_r^*) dx \int_c^d h(y, Y_l) h(y, Y_s^*) dy.$$

Безпосереднє обчислення всіх інтегралів в (4), (5) при великій кількості вузлів сіток за допомогою стандартних систем комп'ютерної математики (наприклад, Mathcad) може вимагати дуже багато часу порівняно з часом розв'язання самої системи (3). В цій роботі пропонується метод, за яким отримуються явні формули для коефіцієнтів  $a_{i,j,r,s}$  і  $b_{r,s}$ . Для обчислення інтегралів (4), (5) скористаємося формулою Сімпсона за кожною змінною. Зазначимо, що підінтегральна функція в кожному з інтегралів є кусково-квадратичною функцією, відмінною від нуля лише на певному інтервалі (залежно від індекса функцій, що є складовими підінтегральної функції). Враховуючи означення функції  $h(x, X_i)$ , робимо висновок, що

$I_{i,r} = \int_a^b h(x, X_i)h(x, X_r)dx = 0$ , якщо  $|i-r| > 1$ , тобто відмінними від нуля є лише три інтеграли  $I_{i,i-1}, I_{i,i}, I_{i,i+1}$  ( $r = i-1, i, i+1$ ), які і підлягають обчисленню.

Безпосередні обчислення дають наступні результати:

$$I_{i,i-1} = \int_{x_{i-1}}^{x_i} \frac{x-x_{i-1}}{x_i-x_{i-1}} \frac{x-x_i}{x_{i-1}-x_i} dx = \frac{x_i-x_{i-1}}{6},$$

$$I_{i,i} = \int_{x_{i-1}}^{x_i} \left( \frac{x-x_{i-1}}{x_i-x_{i-1}} \right)^2 dx + \int_{x_i}^{x_{i+1}} \left( \frac{x-x_{i+1}}{x_i-x_{i+1}} \right)^2 dx = \frac{x_{i+1}-x_{i-1}}{3},$$

$$I_{i,i+1} = \int_{x_i}^{x_{i+1}} \frac{x_{i+1}-x}{x_{i+1}-x_i} \frac{x-x_i}{x_{i+1}-x_i} dx = \frac{x_{i+1}-x_i}{6}.$$

Для інтегралів, що містяться в формулі (5) з урахуванням двох різних сіток отримуюмо аналогічні результати при конкретному співвідношенні між кроками сіток, якщо вони сталі. Наведені результати стосуються функцій двох змінних, але, зрозумілим чином вони переносяться на випадок функцій однієї змінної.

**Приклад.** Описана вище методика використовувалась [4 – 6] для оптимізації числа горизонтальних та вертикальних ліній при побудові поверхні манекена з використанням інтерлінації функцій та з умови найкращого наближення шуканої поверхні (з меншим числом вказаних ліній) до заданої поверхні за допомогою точок на іншій, більш детальній системі горизонтальних та вертикальних ліній. На рис. 1 зображено оригінальний каркас манекена, а на рис. 2 – наближення цього каркасу з меншою кількістю горизонтальних та вертикальних ліній.

**Висновки.** Таким чином, в даній роботі запропоновані та апробовані формули для обчислення інтегралів від функцій, що є добутком сплайнів від двох змінних (першого степеня по кожній змінній), побудованих на різних

сітках вузлів. Ці формули дозволяють значно прискорити обчислення елементів матриці системи рівнянь в методі найменших квадратів.



Рис.1. Каркас манекена, побудований за допомогою 14 горизонтальних та 23 вертикальних ліній



Рис.2. Каркас манекена, побудований за допомогою 13 горизонтальних та 13 вертикальних ліній

**Список літератури:** 1. Крылов В. И., Канторович Л. В. Приближенные методы высшего анализа. – М.: Физматгиз, 1962.– 708 с. 2. Лоусон Ч., Хенсон Р. Численное решение задач метода наименьших квадратов / Пер. с англ. Х. Д. Икрамова. – М.: Наука. Физматгиз, 1986. – 232 с. 3. Михлин С. Г. Вариационные методы в математической физике. – М.: Наука, 1970. – 512 с. 4. Литвин О. М., Пасічник В. О. Оптимізація горизонтальних перерізів математичної моделі поверхні манекена з використанням інтерлінації функцій. // Доповіді НАН України. – 2004. - №2.– С. 66–71. 5. Литвин О. М., Пасічник В. О. Оптимізація числа вертикальних перерізів поверхні манекена та їх розміщення при математичному моделюванні. // Доповіді НАН України. – 2005. - №6.– С. 63–68. 6. Литвин О. Н., Пасечник В. А. Оптимизация математической модели поверхности трехмерного тела. // Кибернетика и системный анализ. – 2006. - №1.– С. 103–112.

Надійшла до редколегії 30.09.09

**СЕЙЕД МОДЖТАБА ДЖАФАРИ ХЕНДЖАНИ**, аспирант НТУ «ХПИ»,  
**В. П. СЕВЕРИН**, д-р техн. наук, профессор НТУ «ХПИ»

### **ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ ОПТИМИЗАЦИИ СЛЕДЯЩЕГО ПРИВОДА И СИСТЕМ УПРАВЛЕНИЯ ПАРОВОЙ ТУРБИНОЙ АЭС**

У статті представлені модифікації генетичних алгоритмів у якості прямих методів глобальної оптимізації для оптимізації векторних цільових функцій. Розроблена й оптимізована математична модель для слідуючого привода парової турбіни у просторі станів з трьома датчиками. Розроблені моделі автоматизованих систем управління частотою обертання ротора парової турбіни у просторі станів з різноманітними стандартними та інтелектуальними регуляторами. Параметри регуляторів оптимізовані для систем стабілізації частоти обертання.

В статье представлены модификации генетических алгоритмов в качестве прямых методов глобальной оптимизации для оптимизации векторных целевых функций. Разработана и оптимизирована математическая модель для следящего привода паровой турбины в пространстве состояний с тремя датчиками. Разработаны модели автоматизированных систем управления частотой вращения ротора паровой турбины в пространстве состояний с различными стандартными и интеллектуальными регуляторами. Параметры регуляторов оптимизированы для систем стабилизации частоты вращения.

Genetic algorithms modifications as global optimization direct methods are developed for vector objective functions optimization. The mathematical model for steam turbine watching drive in state space with three sensors is developed and optimized. The models of automatic control systems for steam turbine rotor rotation frequency are developed in state space with the different standart and intelligent regulators. For the rotation frequency stabilizing systems are optimized regulators parameters.

**Введение.** Одной из основных инженерных проблем создания систем автоматического управления (САУ) паровыми турбинами является проблема оптимизации их показателей качества [1]. Применяющиеся инженерные методы расчета систем управления обычно основаны на существенном упрощении моделей и применении приближенных косвенных скалярных критериев качества. Самым большим недостатком расчетов систем управления турбинами атомных электростанций является то, что в этой области мало применяются модели с оптимальными интеллектуальными регуляторами. Интеллектуальные регуляторы, основанные на нечеткой логике и искусственных нейронных сетях, позволяют решать более сложные задачи управления сложными объектами, чем стандартные регуляторы.

В последние годы интенсивно ведутся работы по практическому внедрению нечетких регуляторов (НР), нечетких экспертных систем и систем управления в промышленную и непромышленную сферы [2]. При этом по качеству переходных процессов и достижению целей управления НР превосходят традиционные ПИД-регуляторы.

Использование интеллектуальных регуляторов приводит к нелинейным неявно заданным моделям САУ. Для оптимального синтеза параметров таких систем с учетом многоэкстремальности их показателей качества целесообразно использовать методы глобальной оптимизации, в частности, методы прямого поиска и генетические алгоритмы (ГА) [3–6]. Эти методы, обычно применяющиеся для оптимизации скалярных целевых функций, необходимо модифицировать для оптимизации векторных целевых функций, учитывающих все требования к САУ [7–9].

Целью данной статьи является разработка моделей систем стабилизации частоты вращения паровой турбины К-1000-1500/60 и ее следящего привода в пространстве состояний с использованием стандартных и НР, а также анализ возможностей оптимизации показателей качества систем стабилизации частоты ГА. Достижение этой цели позволит сделать САУ паровыми турбинами еще эффективнее с учетом всех предъявляемых к ним конструктивных и технологических требований.

**Возможности ГА и их модификация для оптимального синтеза систем управления.** Предыдущий опыт исследования САУ с линейными регуляторами показал, что критерии оптимизации, как правило, являются одноэкстремальными. Однако для сложных многоконтурных систем управления и систем управления с интеллектуальными регуляторами характерно наряду с глобальным наличие большого числа локальных экстремумов. Локальные экстремумы появляются и при введении ограничений на пространство поиска.

Для решения одноэкстремальных задач оптимизации существует достаточное число алгоритмов градиентных методов и методов прямого поиска. Одним из самых эффективных методов прямого поиска является метод деформируемого многогранника Нелдера-Мида [3].

Применение метода Нелдера-Мида для оптимизации САУ с НР различной структуры не приводит к оптимальному решению. В каждом случае результаты зависят от выбранных координат начальной точки поиска. Из этого следует вывод о многоэкстремальности целевых функций подобных задач, и для их решения требуются методы глобальной оптимизации [4].

В настоящее время наиболее предпочтительными методами многоэкстремальной оптимизации являются ГА, реализующие постулаты теории эволюции и опыта селекции растений и животных [5].

Цель оптимизации систем управления с помощью ГА состоит в том, чтобы найти наилучшее возможное решение задачи оптимизации по одному или нескольким критериям. Чтобы реализовать ГА нужно сначала выбрать подходящую структуру для представления этих решений. В постановке задачи поиска, экземпляр этой структуры данных представляет точку в пространстве поиска всех возможных решений. Чтобы оптимизировать структуру, используя ГА, нужно задать некоторую меру качества для каждой структуры в пространстве поиска. Для этой цели используется функция

приспособленности. При максимизации целевой функции часто она сама выступает в качестве функции приспособленности. Для задач минимизации, целевую функцию следует инвертировать и сместить затем в область положительных значений [11]. Стратегия поиска оптимального решения в ГА опирается на гипотезу селекции: чем выше приспособленность особи, тем выше вероятность того, что у потомков, полученных с её участием, признаки, определяющие приспособленность, будут выражены ещё сильнее [6].

Если принять, что каждая особь популяции является точкой в координатном пространстве оптимизационной задачи  $X_i[x_{i1}, x_{i2}, \dots, x_{in}]$ , а приспособленность особи соответствующим значением функции цели  $f(X_i)$ , то популяцию особей можно рассматривать как множество координатных точек в пространстве, а процесс эволюции — как движение этих точек в сторону улучшения значений целевой функции.

Следует отметить, что классический ГА находит глобальный экстремум в вероятностном смысле и эта вероятность зависит от числа особей в популяции. Как показали исследования, при оптимизации сложных многоконтурных и многосвязных систем управления с ПИД-регуляторами и аналогичных систем с интеллектуальными регуляторами ГА с достаточно высокой вероятностью находят глобальный экстремум. В частности, этот факт установлен для диплоидной версии ГА. Однако вычисление целевой функции, связанной с показателями качества системы управления на интервале времени переходного процесса в системе, требует значительных вычислительных ресурсов, что существенно сказывается на общем времени работы ГА.

Особенностью ГА как метода глобального поиска является то, что ни один из ГА (кроссинговер, мутация, инверсия) в процессе генерирования потомков не опирается на знание локального рельефа поверхности отклика для функции цели. Формирование потомков генетическими операторами происходит случайным образом, и поэтому нет гарантии, что найденные решения будут лучше родительских. Следовательно, в процессе эволюции встречается достаточно большое число «неудачных» потомков, которые увеличивают число обращений к функции цели и увеличивают, тем самым, время поиска глобального экстремума.

Кроме того, ГА находят оптимальное решение только внутри заданного диапазона поиска. Поэтому диапазоны поиска и число особей в популяции приходится задавать с большим запасом, что также увеличивает время решения.

Перечисленные особенности сдерживают широкое применение ГА в инженерной практике. Однако потребность в таких алгоритмах для решения прикладных задач сравнительно небольшой размерности постоянно растет, особенно в связи с намечающейся тенденцией внедрения в системы управления нечетких и нейросетевых технологий.

Для оптимального синтеза САУ целесообразно использовать векторные целевые функции, учитывающие весь комплекс требований к системе в порядке их предпочтения [9]. Поскольку ГА в основном используются для оптимизации скалярных целевых функций, то для оптимизации векторных целевых функций эти алгоритмы модифицированы с использованием операций сравнения значений векторной функции [9].

**Моделирование и оптимизация следящего привода.** Электрогидравлический следящий привод является основным исполнительным звеном САУ частотой вращения паровых турбин (САУЧ). Разработаем математическую модель следящего привода, предназначенную для оптимизации его показателей качества.

Принципиальная схема электрогидравлического следящего привода на рис. 1 включает электрогидравлический преобразователь (ЭГП), отсечной золотник (ОЗ), сервомотор (СМ), датчики положения (ДП), электронную часть (ЭЧ). По ихним моделям получены модели схем привода с различным числом датчиков.

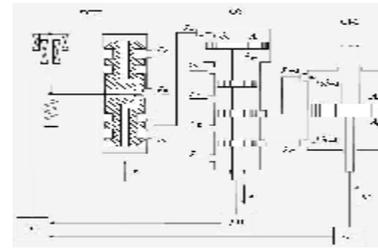


Рис. 1. Принципиальная схема следящего привода

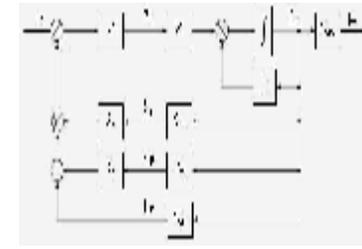


Рис. 2. Структурная схема следящего привода с тремя датчиками

Структурная схема следящего привода с тремя датчиками представлена на рис. 2. Схема включает сумматоры, пропорциональное звено с коэффициентом  $K$ , модель последовательного соединения ЭГП, ОЗ и СМ, датчик положения золотника ЭГП, датчик положения ОЗ, датчик положения СМ, три отрицательные обратные связи (ООС) по положению золотника ЭГП, ОЗ и СМ с коэффициентами  $k_y$ ,  $k_s$  и 1. На вход ЭГП поступает сигнал

$$\xi_u = K(\mu_s - k_y \xi_p - k_s \sigma_p - \mu_p).$$

Для надёжности САУЧ следящий привод должен иметь значительный запас устойчивости, быть чувствительным по отношению к управляющим сигналам и быстродействующим. Рассмотрим оптимизацию показателей качества следящего привода модифицированными ГА.

Для оптимизации привода сформируем вектор из его переменных параметров  $x = (K, k_y, k_s)$ . Наложим на значения варьируемых параметров ограничения  $a_i \leq x_i \leq b_i$ ,  $a_i = 0$ ,  $b_i = 100$ ,  $i = \overline{1, p}$  и сформируем для них штрафную функцию

$$S(x) = \sum_{i=1}^p [\max\{0, a_i - x_i\} + \max\{0, x_i - b_i\}].$$

По штрафной функции ограничений  $S(x)$ , штрафной функции необходимых условий устойчивости  $P(x)$ , коэффициентам первого столбца таблицы Рауса  $\rho_k(x)$ ,  $k = \overline{2, n-1}$ , максимальному отклонению  $\sigma(x)$ , показателю колебательности  $\zeta(x)$ , предельным значениям отклонения и показателя колебательности  $\sigma_m$  и  $\zeta_m$ , времени регулирования  $\tau(x)$  и областям уровней  $H_k$ ,  $k = \overline{0, n+2}$  сформируем учитывающую все ограничения на параметры системы векторную целевую функцию

$$F(x) = \begin{cases} (0; S(x)), & x \in H_0, \\ (1; P(x)), & x \in H_1, \\ (k; -\rho_k(x)), & x \in H_k, k = \overline{2, n-1}, \\ (n; \sigma(x) - \sigma_m), & x \in H_n, \\ (n+1; \zeta(x) - \zeta_m), & x \in H_{n+1}, \\ (n+2; \tau(x)), & x \in H_{n+2}. \end{cases}$$

Первая проекция — функция уровня соответствует числу выполненных ограничений, а вторая проекция функция штрафа определяет штраф нарушения ограничения [9]. Эта функция в приоритетном порядке учитывает непосредственные ограничения на параметры системы, необходимые и достаточные условия устойчивости, ограничения прямых показателей качества, требование минимальности для времени регулирования системы управления. Для оптимизации этой векторной целевой функции были модифицированы ГА с использованием операций сравнения значений векторной функции [9]. Результаты оптимизации привода с применяемыми критериями качества — улучшенной интегральной квадратичной оценки и прямых показателей качества представлены в таблице 1. Для номера вычислительных экспериментов  $N$  приведены найденные модифицированным ГА оптимальные значения варьируемых параметров  $K^*$ ,  $k_y^*$ ,  $k_s^*$  и время регулирования  $t_c^*$ .

Результаты оптимизации следящего привода

$N$	$K^*$	$k_y^*$	$k_s^*$	$t_c^*, c$
1	2,6	0,45	0,75	0,080
2	1,8	0,60	0,90	0,112
3	0,7	0,68	1,26	0,161
4	2,4	0,44	0,77	0,082
5	3,0	0,46	0,69	0,076
6	1,5	0,57	0,98	0,118
7	2,6	0,45	0,75	0,080
8	1,8	0,60	0,90	0,112

Эти результаты, полученные ГА, подтвердили результаты, ранее полученные другими методами оптимизации.

Для  $N = 6$ , изменения переменных состояния оптимального следящего привода, полученные при входном ступенчатом воздействии  $\mu_s = 1(t)$  показали, что все процессы плавные и быстро устанавливаются без существенных колебаний [10]. Координаты тока  $\xi_i$ , перемещения управляющей катушки  $\xi(x)$  и золотника ЭГП  $\xi_y$  из нулевых начальных значений сначала увеличиваются, затем уменьшаются, меняют знак и стремятся к нулевым конечным значениям. Координаты напряжения  $\xi_u$ , ОЗ  $\sigma_s$  и СМ  $\mu_m$  положительны,  $\xi_u$  и  $\sigma_s$  стремятся к нулевым конечным значениям, а  $\mu_m$  — стремится к 1.

Таким образом, оптимизация показателей качества следящего привода позволила обеспечить высокое качество протекающих в нем процессов.

**Моделирование и оптимизация систем управления паровой турбиной.** Рассмотрим построение математических моделей САУ паровой турбиной К-1000-60/1500 в пространстве состояний.

Принципиальная схема паровой турбины представлена на рис. 3 и включает клапан регулирования турбины (КРТ), цилиндр высокого давления (ЦВД), объем перед ЦВД, объем в сепараторе-перегревателе, клапан сепаратора-перегревателя (КСП), объем за клапаном сепаратора-перегревателя, цилиндр среднего давления (ЦСД), цилиндр низкого давления (ЦНД), объем перед ЦНД.

На основании системы дифференциальных уравнений следящего привода, парового тракта и ротора паровой турбины получена математическая модель турбины как объекта управления [9]. САУЧ предназначена для стабилизации частоты вращения ротора паровой турбины. Самым тяжелым испытанием САУЧ является сброс номинальной нагрузки паровой турбины и переход к режиму холостого хода. По разработанной

модели паровой турбины К-1000-60/1500 построим модели САУЧ для сброса нагрузки при различных законах управления с различными регуляторами.

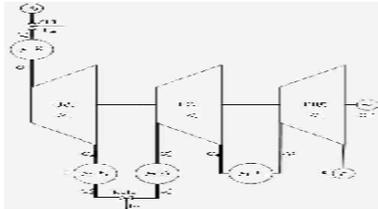


Рис. 3. Принципиальная схема паровой турбины К-1000-60/1500

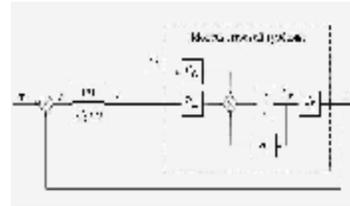


Рис. 4. Структурная схема САУЧ частотой вращения

Структурная схема предложенной модели реальной САУЧ представлена на рис. 4 и включает сумматор, регулятор частоты, модель паровой турбины и ООС [7]. Сигнал ошибки подается на вход стандартного регулятора частоты с передаточной функцией  $W_c(t)$ , формирующий управляющее воздействие  $u$ . Это воздействие и возмущающий сигнал  $v_s$  поступают в модель паровой турбины и изменяют вектор его состояния  $X_T$ . Координата частоты  $\varphi$  ООС подается на сумматор и совместно с задающим воздействием  $\varphi_s$  формирует сигнал ошибки  $\epsilon$  [8]. Входным возмущающим воздействием САУЧ является сигнал изменения мощности  $v_s$ , выходной переменной — частота паровой турбины  $\varphi$ . На основании модели паровой турбины построены математические модели ее систем управления с различными регуляторами [9]. Для оптимизации параметров САУЧ сформируем вектор  $x$  из варьируемых параметров ПИД-регулятора частоты  $K_p$ ,  $\lambda_I$  и  $\lambda_D$ .

Определим модели САУЧ при входном возмущающем воздействии  $v_s = -1(t)$ , соответствующему переходу паровой турбины из режима номинальной мощности  $v = 0$  в режим холостого хода, и выходной координате  $y$  отклонения частоты в процентах:

$$\frac{dX_c(x,t)}{dt} = f(x, X_c(x,t), v_s), \quad y(x,t) = 100 \cdot C_c X_c(x,t),$$

где индекс  $C$  соответствует типу регулятора П, И, Д, ПИ, ПД, ИД или ПИД. САУЧ с И, Д и ИД-регуляторами частоты оказались неэффективными, так как их оптимальные процессы имеют отклонения частоты, превышающие 12%. Результаты оптимизации модифицированными ГА векторной целевой функции представлены в таблице 2 и на рис. 5. Для статических САУЧ с П и

ПД регуляторами  $y(\infty) = 1,0$ , а для астатических САУЧ с ПИ и ПИД-регуляторами  $y(\infty) = 0$ . В первых четырех экспериментах принято допустимое значение размаха колебаний частоты  $\zeta_m = 1$ , а в последних двух  $\zeta_m = 0,5$ . Для  $N = 5$  задано время интегрирования  $T_f = 50$  с, а в остальных экспериментах  $T_f = 20$  с.

Во всех экспериментах оптимальное значение коэффициента пропорциональной составляющей ошибки  $K_p^*$  достигло значения верхнего ограничения  $b_i = 100$ , оптимальные значения  $\lambda_I^*$  и  $\lambda_D^*$  находятся внутри допустимой области изменения параметров. При сбросе нагрузки наилучшим типом регулятора частоты является ПИД-регулятор, обеспечивающий самый быстрый переходный процесс с наименьшим максимальным отклонением частоты и с самым быстрым восстановлением ее номинального значения.

Таблица 2

Результаты оптимизации САУЧ

$N$	$\lambda_I^*$	$\lambda_D^*$	$F_1^*$	$F_2^*$	$\sigma^*$	$\zeta^*$	$t_c^*, c$
1	—	—	20	2,0	3,6	3,0	3,56
2	27	—	22	0,3	3,5	1,0	5,19
3	—	1,1	21	1,7	3,5	2,7	3,16
4	31	1,1	23	0,2	3,4	0,8	3,50
5	10	—	22	0,4	3,6	0,5	19,42
6	23	1,1	23	0,3	3,4	0,5	5,04

На рис. 6 представлены изменения некоторых переменных состояния оптимальной САУЧ с ПИД-регулятором, полученные при входном ступенчатом воздействии  $v_s = -1(t)$ .

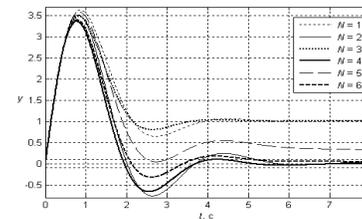


Рис. 5. Изменение частоты при различных регуляторах

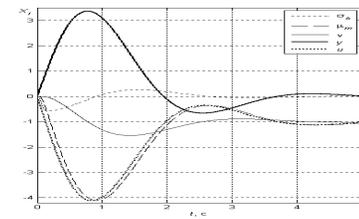


Рис. 6. Изменение переменных состояния в оптимальной САУЧ

Координата ОЗ  $\sigma_s(t)$  изменяется в небольших пределах и устанавливается в нулевое значение. Координата СМ  $\mu_m(t)$ , уменьшаясь, принимает значительные отрицательные значения и переходит к значению  $-1$ . Координата мощности турбины  $v(t)$  из начального нулевого значения номинальной мощности после нескольких быстро затухающих колебаний стремится к значению  $-1$ , соответствующему холостому ходу паровой турбины. Координата отклонения частоты  $y(t)$  из начального нулевого значения после затухающих колебаний возвращается на нулевой номинальный уровень. Сигнал ПИД-регулятора  $u(t)$  уменьшается, а затем после колебаний устанавливается на значении  $-1$ . Выходная координата электрогидравлического следящего привода  $\mu_m(t)$  повторяет с небольшим запаздыванием изменение сигнала регулятора частоты  $u(t)$ . Таким образом, в данной САУЧ с ПИД-регулятором электрогидравлический следящий привод и САУЧ удовлетворяют предъявляемым к ним требованиям.

В отличие от рассмотренных регуляторов нечеткий ПИД-регулятор позволяет организовать более гибкое регулирование по ПИД закону с автоматическим вычислением настроек регулятора для объектов с пропорциональным исполнительным механизмом. Принцип действия НР отличается от классических тем, что на вход объекта, наряду с сигналом регулятора, подается дополнительный пробный синусоидальный сигнал малой амплитуды. По амплитуде и фазе гармонической составляющей в выходном сигнале объекта осуществляется расчет настроек регулятора.

Использование НР позволит лучше автоматизировать процесс настройки контуров регулирования, а также отказаться от использования обычных методов идентификации динамики объекта и расчета оптимальных параметров настройки регуляторов. Как показывает практика, адаптивные регуляторы позволяют экономить до 15% сырья и энергоресурсов по сравнению с ручным управлением или порядка 5% по сравнению с неоптимально настроенным классическим ПИД регулятором [2]. Кроме того, применение адаптивного регулирования ведет к сокращению сроков и стоимости пуско-наладочных работ.

Эффективное совмещение методов теории управления, теории нечетких множеств и нечеткой логики позволяет формировать модели сложно формализуемых процессов управления, а также обеспечивать формирование управляющих сигналов адекватно процессам принятия решений высококвалифицированным специалистом-экспертом [6,11].

Нечеткие системы особенно эффективны в сложных нелинейных процессах АЭС с параметрическими неопределенностями, для которых стандартные подходы синтеза САУ неприменимы. В то же время при синтезе нечетких регуляторов все чаще наблюдается совмещение принципов

нечеткой логики и традиционной теории управления. Применение такого подхода требует анализа устойчивости систем с нечеткими регуляторами.

**Выводы.** Из проведенных исследований вытекают следующие выводы.

1. На основании моделей электрогидравлического преобразователя, отсечного золотника, сервомотора и датчиков положения разработана математическая модель в пространстве состояний следящего привода паровой турбины с тремя датчиками. Оптимизация параметров этой модели обеспечивает высокое качество монотонного процесса в следящем приводе при оптимальном значении коэффициента усиления.

2. Разработаны математические модели систем стабилизации частоты вращения ротора паровой турбины в пространстве состояний с различными стандартными и нечеткими регуляторами частоты. С использованием модифицированных генетических алгоритмов решены задачи оптимизации прямых показателей качества для систем стабилизации частоты вращения. Самым эффективным стандартным типом регулятора частоты при сбросе нагрузки турбины является ПИД-регулятор, который обеспечивает наиболее быстрый переходный процесс с наименьшим отклонением частоты. Проанализированы возможности нечетких ПИД-регуляторов.

3. Для оптимизации векторных целевых функций САУ модифицированы прямые методы безусловной минимизации и генетические алгоритмы, которые позволили найти оптимальные значения показателей.

Вычислительные эксперименты подтвердили высокую эффективность применения модифицированных методов.

**Список литературы:** 1. Фрагин М. С. Регулирование и маслоснабжение паровых турбин: настоящее и ближайшие перспективы. – СПб.: Энерготех, 2005. – 248 с. 2. Прикладные нечеткие системы / Под ред. Т. Тэрано, К. Асаи, М. Сугэно. – М.: Мир, 1993. – 12 с.3. Nelder J. A., Mead R. A simplex method for function minimization, computer journal. – 1964. № 7, P.308-313. 4. Сабанин В. Р., Смирнов Н. И., Репин А. И. Оптимизация настроечных параметров регулирующих устройств в АСР// Сборник трудов конференции Control 2003. – МЭИ, 2003. с 144-148. 5. Goldberg D. E. Genetic algorithms in search optimizations and machine learning. addison.wesly. – 1989. 6. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности // Г. К. Вороновский, К. В. Махотило, С. Н. Петрашев, С. А. Сергеев. Харьков, Основа, 1997. 112 с. 7. Кириллов И. И. Автоматическое регулирование паровых турбин и газотурбинных установок. Л.: Машиностроение, 1988. 447 с. 8. Трояновский Б. М. Паровые турбины. успехи, нерешенные проблемы // Теплоэнергетика. – 1983. №1. С. 6-11. 9. Северин В. П. Моделі і методи оптимізації показників якості систем автоматичного управління енергоблоку атомної електростанції: Автореф. дис... д-ра техн. наук: 05.13.07 / НТУ «ХПШ». Харків, 2007. 36 с. 10. Федянина К. Б., Кучер Е. А., Северин В. П. Оптимизация следящего привода и систем управления паровой турбиной с использованием генетических алгоритмов // Техн. електродинаміка. Тематичний випуск «Силова електроніка та енергоефективність». – 2009. – Ч. 5. – С. 86-91. 11. Джафарі Хенджани Сейед Моджтаба, Северин В. П., Чернай В. Ф. Перспективи применения генетических алгоритмов для синтеза интеллектуальных систем управления энергоблока атомной электростанции // Техн. електродинаміка. Тематичний випуск «Силова електроніка та енергоефективність». – 2009. – Ч. 5. – С. 92-97.

Поступила в редколлегию 30.10.09

**Н. В. КРИВЕНКО**, студент НТУ «ХПИ»

## РЕШЕНИЕ ЗАДАЧИ ВЫБОРА СЦЕНАРИЯ РАЗВИТИЯ ДЛЯ АТОМНОЙ ЭЛЕКТРОСТАНЦИИ

В статье рассматриваются вопросы планирования и выбора сценария развития атомной электростанции. В качестве решения предлагается метод анализа иерархий. Построены иерархия прямого и обратного процесса планирования. В результате расчетов был выбран сценарий развития атомной станции.

The questions of planning and choice of scenario of development of nuclear power plant are examined in the article. As a decision the method of analysis of hierarchies is offered. Built hierarchy of direct and reverse process of planning.

**Постановка задачи.** В работе рассматривается задача выбора сценария развития для атомной электростанции. Задача выбора сценария развития атомной электростанции заключается в выборе одного сценария из множества сценариев  $S\{S_i\}, i = \overline{1, n}$ . Для этого множество сценариев развития оценивают по шкале относительной важности. Для оценивания необходимо привлечь группу экспертов  $E\{E_j\}, j = \overline{1, m}$ . Необходимо построить иерархию прямого и обратного процессов для развития атомной электростанции. Представим задачу в виде иерархии, включающую пять уровней. Иерархия прямого процесса содержит следующие уровни [1]:

Уровень 1. Фокус проблемы. Фокус проблемы – развитие атомной электростанции.

Уровень 2. Факторы. Под факторами понимается какое-либо влияние с окружающей стороны. Выделили такие факторы как: технологические, экономические, политические и экологические.

Уровень 3. Акторы. На этом уровне специалисты по планированию определили, какие акторы влияют на развитие атомной электростанции. Поэтому на этом уровне уместен был вопрос: каково влияние соответственно правительства, руководства АЭС, НАЭК и энерго-ремонтных предприятий по отношению к перспективам развития атомной станции?

Уровень 4. Цели акторов. На этом уровне описаны цели каждого из акторов. Для правительства целями выступает экономическая стабильность, защита окружающей среды и повышение состояния страны. Для НАЭК целями выступает минимизация риска и контроль. Руководство АЭС преследует повышения уровня безопасности и модификацию энергоблоков. Для энерго-ремонтных предприятий – модернизация оборудования.

Уровень 5. Исследовательские сценарии. На этом уровне специалисты по планированию определили три сценария.

Первый сценарий предусматривает строительство двух энергоблоков с реактором типа ВВЭР-1200, второй сценарий предусматривает строительство трёх энергоблоков с реактором типа РWR, третий сценарий – ремонт и замену элементов на имеющихся шести энергоблоках. На рис. 1 представлена иерархия прямого процесса задачи выбора сценария развития АЭС.

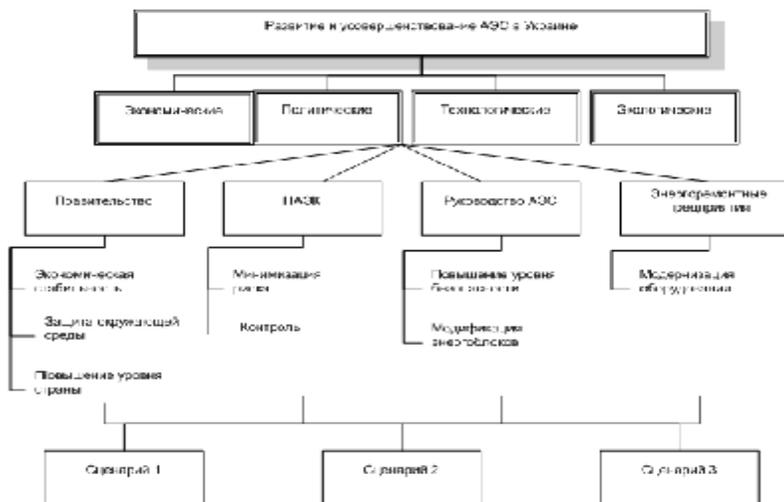


Рис. 1. Иерархия прямого процесса планирования

**Сравнительный анализ и выбор сценария.** Представив задачу в виде иерархии, необходимо элементы второго уровня разместить в матрице и определить какой фактор имеет наибольшее воздействие на развитие атомной станции. Из таблицы 1 видно, что технологический фактор имеет наибольшее воздействие на развитие атомной электростанции.

Таблица 1

Влияние факторов на развитие атомной станции

Развитие АЭС	Технологический	Политический	Экономический	Экологический	Собственный вектор
Технологический	1	4	3	5	0,549
Политический	1/4	1	1/3	1	0,109
Экономический	1/3	3	1	2	0,239
Экологический	1/5	1	1/2	1	0,109
					$I_{\max} = 4.06$

На уровне 3 сравнивается пара акторов относительно степени воздействия на факторы уровня 2. Следовательно, имеем 4 матрицы, приведенные в таблицах 2, 3, 4, 5. Исходя из данных, приведенных в таблицах 2, 3, 4, 5 видно, какое влияние оказывает каждый из акторов на технологию, политику, экономику и экологию.

Таблица 2

Влияние акторов на технологию

Технология	Правительство	НАЭК	Руководство	Энергоремонтные	Собственный вектор
Правительство	1	5	4	6	0.596
НАЭК «Энергоатом»	1/5	1	1	3	0.151
Руководство АЭС	1/4	1	1	5	0.190
Энергоремонтные предприятия	1/4	1/3	1/5	1	0.060 $I_{\max} = 4.17$

Таблица 3

Влияние акторов на политику

Политика	Правительство	НАЭК	Руководство	Энергоремонтные	Собственный вектор
Правительство	1	3	3	1/5	0.220
НАЭК «Энергоатом»	1/3	1	3	1/3	0.139
Руководство АЭС	1/3	1/3	1	1/6	0.065
Энергоремонтные предприятия	5	3	6	1	0.576 $I_{\max} = 4.31$

Таблица 4

Влияние акторов на экономику

Экономику	Правительство	НАЭК	Руководство	Энергоремонтные	Собственный вектор
Правительство	1/4	1/4	1/9	1/7	0.040
НАЭК «Энергоатом»	1	1	1/7	1/7	0.084
Руководство АЭС	9	7	1	1	0.381
Энергоремонтные предприятия	7	7	3	3	0.546 $I_{\max} = 4.40$

Таблица 5

Влияние акторов на экологию

Экология	Правительство	НАЭК	Руководство	Энергоремонтные	Собственный вектор
Правительство	1	1/7	1/3	1/8	0.022
НАЭК «Энергоатом»	7	1	7	1/4	0.105
Руководство АЭС	3	1/7	1	1/7	0.034
Энергоремонтные предприятия	8	4	7	1	0.231 $I_{\max} = 3.67$

Цели каждого из четырёх акторов попарно сравниваются для каждого актора и приведены в таблицах 6, 7, 8. Из таблиц 6, 7, 8 видно какие из целей приоритетнее для каждого актора.

В результате получается собственный вектор, который, по существу, отражает упорядочение и веса целей.

Таблица 6

Сравнение целей для правительства

Правительство	Стабильность	Защита среды	Благосостояние	Собственный вектор
Правительство	1	4	7	0.687
Стабильность	1	4	7	0.687
Защита среды	1/4	1	5	0.243
Благосостояние	1/7	1/5	1	0.069 $I_{\max} = 3.12$

Таблица 7

Сравнение целей для НАЭК «Энергоатом»

НАЭК «Энергоатом»	Минимизация риска	Контроль	Собственный вектор
НАЭК «Энергоатом»	1	1/3	0.250
Минимизация риска	1	1/3	0.250
Контроль	3	1	0.075 $I_{\max} = 2.00$

Таблица 8

Сравнение целей для руководства АЭС

Руководство АЭС	Повышение безопасности	Модификация блока	Собственный вектор
Повышение безопасности	1	3	0.075
Модификация блока	1/3	1	0.250
			$I_{\max} = 2.00$

Следующим этапом было нахождение степени важности акторов относительно факторов, влияющих на развитие атомной станции. Эту оценку провели, умножив, справа матрицы собственных векторов акторов относительно каждого фактора уровня 3 на собственный вектор, который был получен для уровня 2.

$$\begin{bmatrix} 0.596 & 0.22 & 0.040 & 0.022 \\ 0.151 & 0.139 & 0.084 & 0.105 \\ 0.190 & 0.065 & 0.331 & 0.034 \\ 0.06 & 0.576 & 0.546 & 0.231 \end{bmatrix} * \begin{bmatrix} 0.549 \\ 0.106 \\ 0.239 \\ 0.109 \end{bmatrix} = \begin{bmatrix} 0.361 \\ 0.128 \\ 0.194 \\ 0.249 \end{bmatrix} \quad (1)$$

Так как на акторы правительство и энергоремонтные предприятия приходится 61%=(0,361+0,249)\*100, то воздействия на четыре первичных фактора, влияющих на развитие и совершенствование АЭС Украины, было решено использовать только эти два актора для получения весов сценариев.

Необходимо найти важнейшие цели двух акторов: правительства и энергоремонтного предприятия. Для этого нужно умножить собственный вектор целей на соответствующий вес актора.

Для правительства:

$$0.361 * \begin{bmatrix} 0.687 \\ 0.243 \\ 0.069 \end{bmatrix} = \begin{bmatrix} 0.248 \\ 0.087 \\ 0.024 \end{bmatrix} \quad (2)$$

Таким образом, для правительства наиболее влиятельными целями является стабильность и защита окружающей среды.

Для энергоремонтного предприятия – модернизация оборудования. Используя эти три цели и нормализуя их веса, можно получить вектор весов:

$$\begin{bmatrix} 0.248 \\ 0.087 \\ 0.024 \end{bmatrix} = \begin{bmatrix} 0.69 \\ 0.24 \\ 0.07 \end{bmatrix} \quad (3)$$

Этот вектор применяется для получения весов сценариев. Последний этап, который необходим для получения весов сценариев, построение матриц

доминирования относительно каждой цели для трех сценариев, в нашем случае трех целей. Составим матрицу парных сравнений для уровня 4. В таблицах 9, 10, 11 приведены данные влияния каждого из сценариев на экономическую стабильность, на защиту окружающей среды и на модернизацию оборудования.

Таблица 9

Влияние сценариев на экономическую стабильность

Экономическая стабильность	Сценарий 1	Сценарий 2	Сценарий 3	Собственный вектор
Сценарий 1	1	5	4	0.674
Сценарий 2	1/5	1	1/3	0.101
Сценарий 3	1/4	3	1	0.226
				$I_{\max} = 3.086$
				ИС = 0.043
				ОС = 0.074

Таблица 10

Влияние сценариев на защиту окружающей среды

Защита среды	Сценарий 1	Сценарий 2	Сценарий 3	Собственный вектор
Сценарий 1	1	7	1/5	0.233
Сценарий 2	1/7	1	1/8	0.005
Сценарий 3	5	8	1	0.713
				$I_{\max} = 3.247$
				ИС = 0.124
				ОС = 0.213

Таблица 11

Влияние сценариев на модернизацию оборудования

Модернизация оборудования	Сценарий 1	Сценарий 2	Сценарий 3	Собственный вектор
Сценарий 1	1	8	6	0.745
Сценарий 2	1/8	1	1/4	0.065
Сценарий 3	1/6	4	1	0.181
				$I_{\max} = 3.130$
				ИС = 0.068
				ОС = 0.117

Для получения весов сценариев нужно умножить матрицу полученных собственных векторов сценариев на вектор весов трех наиболее важных целей. В результате получаются веса сценариев:

$$\begin{bmatrix} 0.674 & 0.233 & 0.745 \\ 0.101 & 0.005 & 0.065 \\ 0.226 & 0.713 & 0.181 \end{bmatrix} * \begin{bmatrix} 0.69 \\ 0.24 \\ 0.07 \end{bmatrix} = \begin{bmatrix} 0.573 \\ 0.074 \\ 0.339 \end{bmatrix} \quad (4)$$

Таким образом, первый сценарий наиболее выгодный для развития атомной электростанции, за счет наивысшего приоритета. Первый сценарий предусматривает техническое предложение про строительство 2 энергоблоков на атомной электростанции с реактором ВВЭР-1200.

Прямой процесс использовался для определения контрастных сценариев[1]. Иерархию обратного процесса планирования необходимо построить после завершения прямого процесса планирования. Результаты прямого процесса должны быть учтены при построение обратного процесса планирования.

Обратный процесс – нормативная ветвь МАИ. Это определяющий этап, так как в нем идентифицируются политики, которые могут привести к сходимости вероятного и желательного будущего атомной станции[2].

Фокусом иерархии обратного процесса выступает желаемое будущее развитие атомной электростанции.

Уровень 2 обратного процесса включает четырех акторов, которые влияют на реализацию желаемого сценария. В данном примере акторами выступают НАЭК «Энергоатом», Руководство АЭС, Энергоремонтные предприятия, Технологический отдел.

Уровень 3 включает цели организации и их важность относительно ключевых акторов.

Уровень 4 включает возможные программы, которым нужно следовать, чтобы осуществить желаемое будущее. Детализация этого уровня приведена на уровне 5 в виде подпрограмм.

Таким образом, при построении обратного процесса планирования иерархии сводится к программам и подпрограммам, которые необходимы для реализации желаемого результата. Определив приоритеты для элементов каждого уровня, делаем вывод о том, какая из программ и подпрограмм должна быть реализована для достижения желаемого будущего развития атомной электростанции. В данном примере такие программы как: усовершенствование, модификация блока, проверка при эксплуатации имеют наибольший приоритет. В свою очередь программа усовершенствования включает такие подпрограммы, как: программа исключения повреждений, компьютерное моделирование, разработка аналитических методов.

Приоритеты каждого элемента приведены на рис. 2.

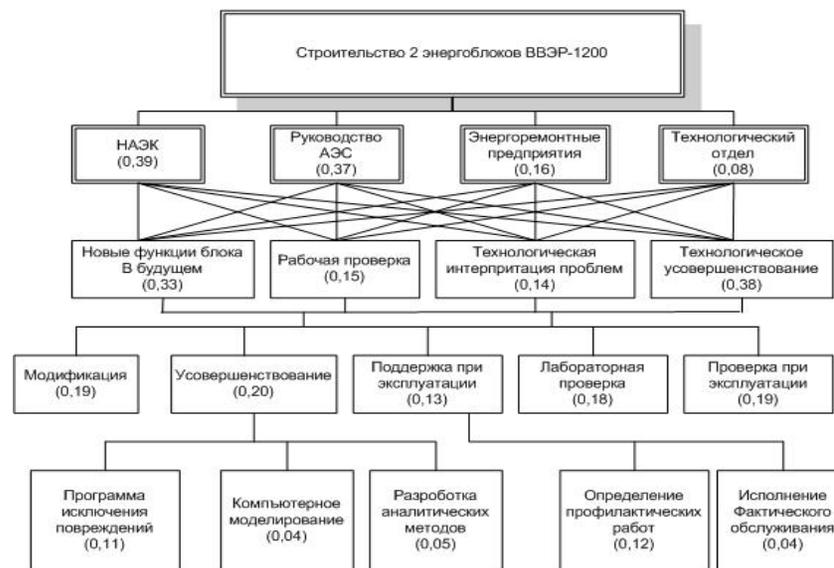


Рис. 2. Иерархия обратного процесса планирования

**Вывод.** В результате применения метода анализа иерархии был получен сценарий развития атомной электростанции, который предусматривает строительство двух энергоблоков.

Метода анализа иерархии успешно применяют во многих областях, в частности: при разработке плана распределения энергии в промышленности; при проектировании транспортной системы; при построении сценариев высшего образования; при выдвижении кандидатов и в процесс их выборов; при определении приоритетов для ведущего научного института в развивающейся стране[3]. К преимуществам МАИ можно отнести то, что при определении иерархии важную роль играют знания лиц, производящих суждения для парных сравнений.

**Список литературы:** 1. Т.Саати, К. Кернс. Аналитическое планирование. – М.: Радио и связь, 1991, 224с. 2. Абакаров А.Ш., Сушков В.А. Программная система поддержки принятия рационального решения. – Электронный научный журнал «ИССЛЕДОВАНИЕ В РОССИИ», №1, 2005 г. 3. Носин В.Д. Принятие решения в многокритериальной среде: количественный подход. – М.: Физматлит, 2002, 176с.

Поступила в редколлегию 15.12.09

**Т. В. НЕСКОРОДЕВА**, канд. техн. наук

### ЛОГИКО-ФОРМАЛЬНЫЕ ПРАВИЛА ФОРМИРОВАНИЯ ТОВАРНОГО ОБЕСПЕЧЕНИЯ ТОРГОВОГО ПРЕДПРИЯТИЯ

У статті виконаний системний аналіз торгівельного підприємства як об'єкта управління, запропонований метод контролю формування товарного забезпечення торгівельного підприємства, виконана формалізація правил структурування договорів на поставку і прецедентів прийняття рішень щодо товарного забезпечення підприємства.

В статье выполнен системный анализ торгового предприятия как объекта управления, предложен метод контроля формирования товарного обеспечения торгового предприятия, выполнена формализация правил структурирования договоров на поставку и прецедентов принятия решений относительно товарного обеспечения предприятия.

In article the system analysis of trade enterprise as object of management is executed, there have been proposed methods of the quality monitoring of commodity maintenance formation of trade enterprise, formalization structurization rules of contracts on delivery and precedents is executed at decision-making concerning commodity maintenance of the enterprise.

**Введение.** Современные обстоятельства, бизнес-среда требуют комплексного усовершенствования инструментария и технологий аудита на всех этапах процесса принятия и реализации управленческих решений. Основным недостатком существующих методик аудита является отсутствие формализации и четкого выделения элементов контрольного процесса. Именно это усложняет внедрение методик контроля в практику хозяйствования предприятий и порядок их реализации. Результат основной деятельности торгового предприятия существенно зависит от решений по товарному обеспечению. Конкуренция крупных торговых компаний приводит к тому, что внутренние аудиторы стремятся повысить эффективность основной (торговой) деятельности за счет усиления контроля за принятием и реализацией управленческих решений. Поэтому актуальной задачей является разработка моделей и методов контроля формирования товарного обеспечения торговых предприятий.

Вопросы организации внутреннего аудита торгового предприятия и проблемы формирования его товарного обеспечения описаны в [1]. Теоретические основы адаптивного управления изложены в [2–5]. Некоторые аспекты разработки ИТ СППР при управлении предприятием приведены в [6]. Модели и информационные технологии поливариантного анализа деятельности предприятия при внешнем аудите изложены в работе [7].

Постановка задачи: разработать модели и метод контроля формирования товарного обеспечения.

В данной работе выполнен системный анализ торгового предприятия как объекта управления; формализация показателей основной деятельности и

показателей соответствующих аудиторской проверке, описаны информационные потоки и переменные, характеризующие показатели договоров на поставку продукции и на основании проведенных исследований выполнена формализация правил и предложен метод контроля принятия решений при формировании товарного обеспечения.

**Основная часть.** Исследование торговой компании как объекта управления позволило выделить в управлении основной (торговой) деятельностью три уровня: правление торговой компании (первый уровень); планово-экономический отдел (второй уровень); отделы реализации готовой продукции и отдел поставок (третий уровень).

Для определения задач контроля принятия решений необходимо определить задачи планирования для каждого из трех уровней. На первом уровне правлением советом директоров решается задача планирования, которая заключается в следующем: определить основные направления развития торговой деятельности в зависимости от сложившейся ситуации на рынке и финансового состояния торговой компании и установить интервалы изменения торговых наценок по  $j$ -му виду группы товаров, таким образом, чтобы максимизировать прибыль от основной деятельности  $P(t)$ . Таким образом, множеством входных переменных  $XPL_1$  для первого уровня управления являются показатели Баланса на начало планового периода  $C_s(t_0)$  ( $s$  – статья Баланса,  $s = \overline{1, S}$ ), оценки экспертов о состоянии рынка по группам товаров, а также результаты работы торговой компании за предыдущие периоды времени, а именно, объемы реализации  $OR_j(t-1)$ , поставок  $OS_j(t-1)$ , торговые наценки  $Tar_j(t-1)$  и прибыль  $P_j(t-1)$  по группам товаров и прибыль по всей торговой компании  $P(t-1)$ :

$$XPL_1 = \left\{ C_s(t_0), s = \overline{1, S}, P(t-1), OS_j(t-1), \right. \\ \left. OR_j(t-1), Tar_j(t-1), j = \overline{1, n_{vid}} \right\}, \quad (1)$$

где  $n_{vid}$  – количество групп товаров.

В качестве выходных переменных  $YPL_1$  первого уровня управления выступают прибыль  $P^{ie}(t)$ , которую нужно получить за плановый период времени (месяц, квартал, год и т.д.), объемы поставок и реализации в денежном измерении по каждой товарной группе  $OS_j^{ie}(t)$ ,  $OR_j^{ie}(t)$  интервалы изменения торговых наценок по  $j$ -й группе товаров  $[Tar_j'^{ie}(t); Tar_j''^{ie}(t)]$ , которые в виде задания поступают на второй уровень управления:

$$YPL_1 = \{P^{i\bar{e}}(t), OS_j^{i\bar{e}}(t), OR_j^{i\bar{e}}(t), [Tar_j^{i\bar{e}}(t); Tar_j^{*i\bar{e}}(t)], j = \overline{1, n_{vid}}\}. \quad (2)$$

На вход второго уровня управления поступают сведения о показателях Баланса на начало планового периода  $C_s(t_0)$  ( $s$  – статья Баланса,  $s = \overline{1, S}$ ), совокупности поступивших предложений поставщиков  $Z$ , действующих договорах поставок  $D1$ , и договорах, по которым истек срок действия  $D2$ , а также значения выходных переменных  $YPL_1$  из первого уровня управления:

$$XPL_2 = \{C_s(t_0), s = \overline{1, S}, Z, D_1, D_2, P^{i\bar{e}}(t), OS_j^{i\bar{e}}(t), OR_j^{i\bar{e}}(t), [Tar_j^{i\bar{e}}(t); Tar_j^{*i\bar{e}}(t)], j = \overline{1, n_{vid}}\}. \quad (3)$$

Задача планирования второго уровня состоит в формировании плана поставок и реализации таким образом, чтобы прибыль от основной деятельности была максимальной и не ниже плановой  $P^{i\bar{e}}(t)$ . Множество выходных переменных второго уровня  $YPL_2$  управления характеризуется плановыми заданиями для отделов поставок: обеспечить поставку  $n_{i(j)}^{i\bar{e}}(t)$  единиц товара  $i$ -го вида ( $i = \overline{1, N(j)}$ ) в  $j$ -й товарной группе, в ценовом интервале  $[Cen_{i(j)}^{i\bar{e}}(t); Cen_{i(j)}^{*i\bar{e}}(t)]$  на сумму не более  $OS_{i(j)}^{i\bar{e}}(t)$  и отделам продаж по группам товаров: реализовать  $nr_{i(j)}^{i\bar{e}}(t)$  единиц товаров  $i$ -го вида  $j$ -й группы на сумму не менее  $OR_{i(j)}^{i\bar{e}}(t)$  с торговой наценкой  $Tar_j^{i\bar{e}}(t) \in [Tar_j^{i\bar{e}}(t); Tar_j^{*i\bar{e}}(t)]$  таким образом, чтобы обеспечить максимальную прибыль  $P_j^{i\bar{e}}(t)$  по данной группе товаров:

$$YPL_2 = \{n_{i(j)}^{i\bar{e}}(t), nr_{i(j)}^{i\bar{e}}(t), [Cen_{i(j)}^{i\bar{e}}(t); Cen_{i(j)}^{*i\bar{e}}(t)], Tar_j^{i\bar{e}}(t), OR_{i(j)}^{i\bar{e}}(t), OS_{i(j)}^{i\bar{e}}(t), P_j^{i\bar{e}}(t), i(j) = \overline{1, N(j)}, j = \overline{1, n_{vid}}\}. \quad (4)$$

Выходные переменные  $YPL_2$  второго уровня являются входными для планирования на третьем уровне:

$$XPL_{3,j} = \{n_{i(j)}^{i\bar{e}}(t), nr_{i(j)}^{i\bar{e}}(t), Cen_{i(j)}^{i\bar{e}}(t), Tar_j^{i\bar{e}}(t), OR_{i(j)}^{i\bar{e}}(t), OS_{i(j)}^{i\bar{e}}(t), P_j^{i\bar{e}}(t)\}, j = \overline{1, n_{vid}} \quad (5)$$

Задача планирования третьего уровня заключается в организации работы менеджеров по поставкам и продавцов-консультантов по  $j$ -й товарной группе таким образом, чтобы обеспечить выполнение планового задания

второго уровня  $n_{i(j)}^{i\bar{e}}(t), nr_{i(j)}^{i\bar{e}}(t), Cen_{i(j)}^{i\bar{e}}(t), Tar_j^{i\bar{e}}(t), OR_{i(j)}^{i\bar{e}}(t), OS_{i(j)}^{i\bar{e}}(t)$  и получить максимальный доход по  $j$ -й товарной группе не ниже  $P_j^{i\bar{e}}(t)$  рассчитанного на втором уровне.

В качестве решения задачи планирования на третьем уровне выступают задания для  $k$ -го менеджера по поставкам ( $k = \overline{1, n_{men}}$ , где  $n_{men}$  – количество менеджеров): количество товаров  $n_{k,i(j)}^{i\bar{e}}(t)$  по  $i(j)$ -му виду товара  $j$ -й товарной группы, договора на поставку которых должен заключить  $k$ -й менеджер, на сумму не более плановой  $OS_{\bar{e},i(j)}^{i\bar{e}}(t)$ , и задания для  $s$ -го продавца-консультанта ( $s = \overline{1, n_r}$ , где  $n_r$  – количество продавцов-консультантов): количество товаров по  $i(j)$ -му виду товара  $j$ -й товарной группы, которые должен реализовать  $s$ -й продавец-консультант  $nr_{s,i(j)}^{i\bar{e}}(t)$ , на сумму не меньше плановой  $OR_{s,i(j)}^{i\bar{e}}(t)$ . Т.е. множеством выходных переменных третьего уровня при планировании основной деятельности торговой компании является

$$YPL_{3,j} = \{n_{k,i(j)}^{i\bar{e}}(t), OS_{\bar{e},i(j)}^{i\bar{e}}(t), k = \overline{1, n_{men}}, nr_{s,i(j)}^{i\bar{e}}(t), OR_{s,i(j)}^{i\bar{e}}(t), s = \overline{1, n_r}, i = \overline{1, N(j)}\}, j = \overline{1, n_{vid}}. \quad (6)$$

В соответствии с разработанным планом менеджеры по поставкам заключают договора на поставку  $d_m \in D_1$  (где  $D_1$  – множество действующих договоров данной торговой компании,  $m$  – номер договора), а продавцы-консультанты реализуют план продаж.

Анализ работы внутренних аудиторов, позволил установить, что цель контроля в плановом режиме – выявление проблем, которые могут стать причиной невыполнения плановых показателей на трех уровнях управления  $\{YPL_1, YPL_2, YPL_{3,j}, j = \overline{1, n_{vid}}\}$  и выработка управляющих воздействий, направленных на выполнение плана.

Анализ деятельности торговых компаний показал, что выявление проблем в плановом режиме осуществляется на основании проверки на наличие прецедентов проблемных ситуаций и аналогий (проблемная ситуация может быть идентифицирована как по отчетным данным так и прогнозным расчетам полученным по результатам экономико-математического моделирования). Последовательность операций контроля разработки решений в режиме планирования «сверху вниз» (от первого уровня к третьему). Сформированная гипотеза о механизме контроля, является основанием для формализации процесса контроля в плановом режиме.

Процесс контроля принятия решений по торговому обеспечению предприятия осуществляется на основании информации по действующим договорам, договорам торговой статистики, а также договорам, которые находятся на стадии согласования.

На основании анализа деятельности торговой компании, нормативно-правовых актов регулирующих заключение договоров были выделены следующие характеристики договоров и определена их структура.

Обозначим множество всех договоров, по которым хранится информация в базе данных ИТ СППР „Аудит” через  $D$ , а договора поставки – через  $d_m, m = \overline{1, M}$ , где  $m$  и  $M$  номер и количество договоров соответственно. Тогда множество договоров имеет вид:  $D = \{d_m, m = \overline{1, M}\}$ . В качестве характеристик договора поставки  $d_m$  выступают:

$$\forall d_m \in D: d_m = (c_f^m, b_{i(j)}, \overline{b_{i(j)}^m}, Cen_{i(j)}^m, \overline{har_{i(j)}^m}, j \in \tilde{J}_m, i \in \tilde{I}_m(j),$$

$$ss_m, r_m, CS_m, data_m, sr_m, \overline{sr_m}, \overline{Dsr_m},$$

$$\overline{b}_m, \tilde{b}_m, \overline{ss}_m, \overline{prb}_m, \overline{nb}_m, prs_m, \overline{men}_k^m), \quad (7)$$

где  $m$  – номер договора,

$c_f$  – поставщик ( $c_f \in C$ , где  $C$  – совокупность поставщиков компании,

$f$  – номер поставщика);

$b$  – объем поставки по договору;

$\overline{b} = (b^1, b^2, \dots, b^{Np})$  – партии поставок;

$Cen$  – цена товара;

$\overline{har} = (har^1, har^2, \dots, har^L)$  – характеристики товара;

$\tilde{J}$  – множество товарных групп, по которым заключен договор;

$\tilde{I}(j)$  – множество товаров в товарной группе, по которым заключен договор;

$ss$  – сумма договора;

$r$  – условия оплаты ( $r \in \{0, 1, 2\}$ ) (0 – без предоплаты, 1 – с частичной предоплатой, 2 – с полной предоплатой);

$CS = \{cs_{u1}, cs_{u2}, u = \overline{1, U}\}$  – условия скидок ( $u$  и  $U$  – номер и количество уровней скидки соответственно,  $cs_{u1}$  – объем скидки,  $cs_{u2}$  – минимальный объем поставки для получения скидки  $u$ -го уровня);

$pr$  – условия поставки ( $pr \in \{0, 1\}$ ;  $pr = 0$ , если расходы на формирование, транспортировку товарных ресурсов несет поставщик, в противном случае  $pr = 1$ );

$data$  – дата заключения договора;

$sr$  – срок договора ( $sr \in K$ , где  $K$  – совокупность сроков заключения договоров);

$\overline{sr} = (sr_1, sr_2, \dots, sr_{Np})$  и  $\overline{\Delta sr} = (\Delta sr_1, \Delta sr_2, \dots, \Delta sr_{Np})$  – сроки поставок и задержек соответственно ( $\{sr_p, p = \overline{1, Np}\} \in [data, data + sr]$ );

$\overline{ss} = (ss^1, ss^2, \dots, ss^{Np})$  – суммы поставок ( $\sum_{p=1}^{Np} ss^p = ss$ );

$\tilde{b}$  – объем выполненных поставок по договору ( $\tilde{b} \leq b$ );

$\overline{ss}$  – объем платежей перечисленных по договору ( $\overline{ss} \leq ss$ );

$prb$  – объем просроченных поставок по договору ( $prb \leq b$ );

$nb$  – объем невыполненных поставок по договору ( $nb \leq b$ );

$prs$  – сумма просроченных выплат по договору ( $prs \leq ss$ );

$men_k$  – менеджер, заключивший договор ( $men_k \in Men$ , где  $Men$  – множество менеджеров торговой компании).

Структура договоров определяется в зависимости от срока их действия и имеет следующий вид:

$$D = D_1 \cup D_2 \cup Z, \quad D_1 \cap D_2 \cap Z = \emptyset, \quad (8)$$

где  $D_1$  – множество действующих договоров,

$D_2$  – множество договоров торговой статистики,

$Z$  – множество договоров на стадии согласования.

Для определения принадлежности договора  $d_m$  одному из множеств  $Z, D_1, D_2$  разработаны следующие логико-формальные правила:

1. Если по договору не наступил срок первой поставки, то договор еще не вступил в силу, и является заявлением на поставку:

$$\forall d_m \in D: (sr_1 > data_0) \Rightarrow d_m \in Z \quad (data_0 - \text{текущая дата}).$$

2. Если по договору закончился срок действия, и нет задолженности торговой компании по расчетам, то он принадлежит множеству  $D_2$  – договоров торговой статистики:  $\forall d_m \in D:$

$$(data_0 < (data_m + sr_m)) \vee (prs = 0) \Rightarrow d_m \in D_2.$$

3. Договор, не являющийся заявлением на поставку и не относящийся к торговой статистике, принадлежит множеству действующих договоров:  $\forall d_m \in D: (d_m \notin D_2) \wedge (d_m \notin Z) \Rightarrow d_m \in D_1$ .

Структура поставщиков определяется в зависимости от их надежности по объемам и срокам поставок:  $C = \bigcup_{kat=1}^{N_{km}} C_{kat}$ , где  $kat_f$  – номер категории поставщика, характеризующий его надежность. В зависимости от номера категории поставщик определяется следующим образом:

$$\text{если } kat_f = \begin{cases} 0, \text{ то поставщик не надежный по поставкам;} \\ 1, \text{ то поставщик не надежный и по срокам и} \\ \quad \text{по объемам поставок;} \\ 3, \text{ то поставщик надежный и по срокам и} \\ \quad \text{по объемам поставок;} \\ 2, \text{ то поставщик среднего уровня надежности.} \end{cases}$$

Категория поставщика определяется на основании следующих трех коэффициентов надежности: выполнения поставок, по объемам поставок, по срокам поставок. Коэффициент надежности выполнения поставок, определяется как отношение суммы объемов невыполненных поставок к общей сумме поставок по всем договорам, заключенным с данным поставщиком:

$$nad_f^1 = \left( \sum_{m \in d(c_f)} \sum_{j \in J} \sum_{i(j)=1}^{N(j)} nb_{i(j)}^m Cen_{i(j)}^m \right) / \sum_{m \in d(c_f)} ss_m.$$

Коэффициент надежности по объемам поставок определяется как отношение суммы просроченных поставок к общей сумме поставок по всем договорам, заключенным с данным поставщиком:

$$nad_f^2 = \left( \sum_{m \in d(c_f)} \sum_{j \in J} \sum_{i(j)=1}^{N(j)} prb_{i(j)}^m Cen_{i(j)}^m \right) / \sum_{m \in d(c_f)} ss_m.$$

Коэффициент надежности по срокам поставок определяется как отношение суммарного времени задержек поставок к сроку действия по всем договорам, заключенным с данным поставщиком:

$$nad_f^3 = \left( \sum_{m \in d(c_f)} \sum_{p=1}^{N_p} \Delta sr_p^m \right) / \sum_{m \in d(c_f)} sr_m.$$

Тогда номер категории поставщика, характеризующий его надежность определяется по логико-формальным правилам:

$$kat_f = \begin{cases} 0, \text{ если } (0,2 \leq nad_f^1 \leq 1); \\ 1, \text{ если } (0,6 \leq nad_f^2 \leq 1) \wedge (0,6 \leq nad_f^3 \leq 1); \\ 3, \text{ если } (0 \leq nad_f^2 \leq 0,3) \wedge (0 \leq nad_f^3 \leq 0,3); \\ 2, \text{ в противном случае.} \end{cases}$$

Пределы изменения коэффициента надежности в каждой группе и количество категорий определяется торговой компанией на основании политики компании по отношению к поставщикам (которая базируется на анализе рынка предложений по данной группе товаров).

Структура товаров в каждой товарной группе определяется по интегрированному критерию цена-качество. Для определения принадлежности видов товара к товарным группам разработаны логико-формальные правила.

1. Если качество и цена товара высокие, то товар принадлежит к множеству элитных товаров  $I_1(j)$  в товарной группе  $j$ :

$$\forall i \in I(j): (har_{i(j)}^1 = 1) \wedge (Cen_{i(j)} > Cen'_{i(j)}) \Rightarrow i \in I_1(j),$$

где  $Cen'_{i(j)} < Cen''_{i(j)}$ .

2. Если качество и цена товара низкие, то товар принадлежит к социальной группе товаров:

$$\forall i \in I(j): (har_{i(j)}^1 = 3) \wedge (Cen_{i(j)} < Cen'_{i(j)}) \Rightarrow i \in I_3,$$

где  $Cen'_{i(j)} > Cen''_{i(j)}$ .

3. Если товар не является ни социальным, ни элитным, то принадлежит к множеству товаров «среднего класса»:

$$\forall i \in I(j): (i \notin I_1) \wedge (i \notin I_3) \Rightarrow i \in I_2.$$

Таким образом, структура номенклатуры товарооборота торговой компании имеет вид:

$$\bigcup_{j \in J} I(j) = \bigcup_{j \in J} I_1(j) \cup I_2(j) \cup I_3(j).$$

Структура товарных групп определяется в зависимости от скорости их реализации. Для определения принадлежности товарной группы разработаны логико-формальные правила.

1. Если средний срок реализации товаров данной товарной группы не превосходит некоторого срока (определенного торговой компанией)  $srr'$ , то товарная группа принадлежит к множеству быстро реализуемых товарных групп:

$$\forall j \in J : (\overline{srr}_j \leq srr') \Rightarrow j \in J_2.$$

2. Если средний срок реализации товаров данной товарной группы превосходит некоторый срок (определенный торговой компанией)  $srr''$  ( $srr'' < srr'$ ), то товарная группа принадлежит к множеству медленно реализуемых товарных групп:

$$\forall j \in J : (\overline{srr}_j > srr'') \Rightarrow j \in J_2.$$

3. Если товарная группа не является ни быстро реализуемой, ни медленно, то принадлежит к множеству товарных групп «среднего срока реализации»:

$$\forall j \in J : (j \notin J_1) \wedge (j \notin J_3) \Rightarrow j \in J_2.$$

Следовательно, структура товарных групп в зависимости от скорости их реализации имеет вид:  $J = J_1 \cup J_2 \cup J_3$ .

На основании информации по договорам торговой компании на поставку продукции аудитор выявляет проблемы, которые могут быть причиной невыполнения плановых показателей поставок, определенных в формулах (2), (4), (6). На основании анализа деятельности внутренних аудиторов торговой компании были выделены и формализованы прецеденты при формировании товарного обеспечения на трех уровнях управления, начиная с первого. Для их формализации введем индекс проблемных договоров, который определяется как соотношение суммы поставок по проблемным договорам  $l$ -го вида к плановому объему поставок:

$$h_l = \frac{\sum_{(m \in M_l) \wedge (sr_p \in [t_0, t_0+t])} ss_m(sr_p)}{\sum_{j=1}^{n_{ij}} OS_j^{ip}(t)}, \quad l = \overline{1, L},$$

где  $M_l$  – множество проблемных договоров.

На первом уровне были выделены следующие проблемы.

- 1–3. Индекс проблемных договоров  $l$ -го вида за плановый период  $[t_0, t+t]$  превышает критический уровень  $ur_l$ :

$$h_l > ur_l, \quad l = \overline{1, 3},$$

где  $M_1$  – множество проблемных договоров первого вида (с поставщиками, которые имеют прецеденты нарушения сроков и объемов поставок), которое определяется по правилу:  $if \ c_f \in C \setminus (C_3 \cup C_2) \Rightarrow d_m(c_f) \in M_1$ ,

$M_2$  – множество проблемных договоров второго вида (с поставщиками на условиях предоплаты за товар с низкой скоростью реализации), которое определяется по правилу:  $if \ (j \in J_2) \vee ((r=1) \wedge (r=2)) \Rightarrow d_m(j, r) \in M_2$ ,

$M_3$  – множество проблемных договоров третьего вида (без учета затрат на формирование, транспортировку и хранение товарных ресурсов), которое определяется по правилу:  $if \ pr=1 \Rightarrow d_m(pr) \in M_3$ .

4. Объем проблемных договоров трех видов  $M_4$  ( $M_4 = \bigcup_{l=1}^L M_l$ )

превышает критический уровень  $ur_4$ :

$$\sum_{l=1}^3 h_l > ur_4.$$

5. Составление договоров относительно приобретения товаров с поставщиками без определения оптимального соотношения ассортимента, качества и цен товарной продукции (объемы элитных поставок больше критического уровня  $ur_5$  или объемы социальных поставок ниже критического уровня  $ur_6$ ):

$$(h_5 > ur_5) \vee (h_6 < ur_6),$$

где коэффициенты  $h_5$  и  $h_6$ , определяются равенством (6) для множества  $M_5$  договоров на поставку элитных товаров, и для множества  $M_6$  договоров на поставку социальной группы соответственно, которые, в свою очередь, определяются по следующим логико-формальным правилам:

$$M_5 \subseteq Z \cup D_1 : \forall d_m(i) \in M_4 \Rightarrow i \in I_1(j);$$

$$M_6 \subseteq Z \cup D_1 : \forall d_m(i) \in M_6 \Rightarrow i \in I_3(j).$$

На втором уровне были выделены следующие проблемы.

1. Соотношение объемов поставок  $i(j)$ -го вида продукции по проблемным договорам первого вида к плановому объему поставок превышает критический уровень  $ur_{i(j)}^1$ ,  $i \in I_{pr}^1(j)$ ,  $j \in J_{pr}^1$ :

$$\frac{\sum_{(m \in M_1) \wedge (sr_p \in [t_0, t_0+t])} b_{i(j)}^m(sr_p)}{n_{i(j)}^{\bar{i}\bar{e}}(t)} > ur_{i(j)}^1.$$

2–4. Индекс проблемных договоров  $l$ -го вида в  $j$ -й товарной группе превышает критический уровень  $ur_j^l$ ,  $j \in J_{pr}^l$ :

$$h_j^l > ur_j^l, \quad l = \overline{1,3},$$

$$\text{где } h_j^l = \frac{\sum_{i=1}^{N(j)} \sum_{(m \in M_1) \wedge (sr_p \in [t_0, t_0+t])} b_{i(j)}^m(sr_p) Cen_{i(j)}^m}{OS_j^{\bar{i}\bar{e}}(t)}.$$

5. Объем проблемных договоров трех видов  $j$ -й товарной группе превышают критический уровень  $ur_j^4$ ,  $j \in J_{pr}^4$ :

$$\sum_{l=1}^3 h_j^l > ur_j^4.$$

6. Составление договоров в  $j$ -й товарной группе без определения оптимального соотношения ассортимента, качества и цен товара:

$$(h_j^5 > ur_j^5) \vee (h_j^6 < ur_j^6).$$

На третьем уровне были выделены следующие проблемы.

1. Соотношение объемов поставок  $i(j)$ -го вида продукции по проблемным договорам первого вида, заключенных  $k$ -м менеджером к его плановому заданию  $n_{k,i(j)}^{\bar{i}\bar{e}}(t)$  превышает критический уровень  $ur_{k,i(j)}^1$ ,  $i \in I_{pr}^1(j)$ ,  $j \in J_{pr}^1$ ,  $k \in \{1, n_{men}\}$ :

$$\frac{\sum_{d_m(men_k), (m \in M_1) \wedge (sr_p \in [t_0, t_0+t])} b_{i(j)}^m(sr_p)}{n_{k,i(j)}^{\bar{i}\bar{e}}(t)} > ur_{k,i(j)}^1.$$

2–4. Индекс проблемных договоров  $l$ -го вида в  $j$ -й товарной группе, заключенных  $k$ -м менеджером превышает критический уровень  $ur_{k,j}^l$ ,  $j \in J_{pr}^l$ ,  $l = \overline{1,3}$ :

$$h_{k,j}^l > ur_{k,j}^l,$$

$$\text{где } h_{k,j}^l = \frac{\sum_{i=1}^{N(j)} \sum_{d_m(men_k), (m \in M_1) \wedge (sr_p \in [t_0, t_0+t])} b_{i(j)}^m(sr_p) Cen_{i(j)}^m}{\sum_{i=1}^{N(j)} OS_{k,i(j)}^{\bar{i}\bar{e}}(t)}.$$

5. Индекс проблемных договоров трех видов в  $j$ -й товарной группе, заключенных  $k$ -м менеджером к его плановому заданию превышают критический уровень  $ur_{k,j}^4$ :

$$\sum_{l=1}^3 h_{k,j}^l > ur_{k,j}^4.$$

Значения критических уровней оцениваются на основании статистических данных работы торгового предприятия за предыдущие плановые периоды, или на основании экспертных оценок.

**Заключение.** В работе структурированы договора на поставку, разработаны модели и предложен метод контроля формирования товарного обеспечения торгового предприятия по прецедентам, который предусматривает возможность выявлять проблемы в товарном обеспечении предприятия на этапе принятия решений.

Дальнейшее развитие методики внутреннего аудита на базе данных результатов предполагает выявлять проблемы реализации продукции, выполнения плановых показателей прибыли, платежеспособности и финансовой устойчивости предприятия. Планируется исследовать взаимосвязи между проблемами на каждом уровне и между уровнями.

**Список литературы:** 1. Буцез В. В. Організація системи внутрішнього контролю комерційної організації. – М.: «Спит», 2000. – 320 с. 2. Цыпкин Я. З. Адаптация и обучение в автоматических системах. – М.: Наука, 1968. – 252 с. 3. Жданов А. А. Метод автономного адаптивного управления // Известия Академии наук. Теория и системы управления, 1999. – № 5. – С. 127–134. 4. Карнов Л. Е., Юдин В. Н. Методы добычи данных при построении локальной метрики в системах вывода по прецедентам // ИСП РАН, препринт № 18, М. – 2006. – С. 322–334. 5. Кузьменко В. В., Гришин Д. В. Теоретические аспекты функционирования адаптивной системы управления предприятием // Вестник СевКавГТУ, Серия "Экономика", 2003. – № 2 (10). – С. 25–33. 6. Єршова О. Л. Моделі, методи і засоби інформаційної технології прийняття управлінських рішень у соціально-економічних системах: автореф. дис. на здобуття наукового ступеня к.е.н.: спец. 08.03.02, К. – 2000. – 19 с. 7. Нескородєва Т. В. Моделі та інформаційні технології поліваріантного аналізу діяльності підприємства: автореф. дис. на здобуття наук. ступеня канд. техн. наук: спеціальність 05.13.06 „Інформаційні технології” / Т. В. Нескородєва – Харків, 2008. – 19 с.

Поступила в редакцію 18.11.09

**С. С. КУЦ**, студент НТУ «ХПИ»,  
**А. С. КУЦЕНКО**, д-р тех. наук, профессор НТУ «ХПИ»,  
**И. И. МАРЧЕНКО**, ассистент НТУ «ХПИ»

### КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ФОРМИРОВАНИЯ ПОВЕРХНОСТИ ТОНКИХ ПЛЁНОК КИНЕТИЧЕСКИМ МЕТОДОМ МОНТЕ-КАРЛО

У роботі розроблена модель осадження тонких плівок, яка заснована на методі кінетичного Монте-Карло. Запропонований алгоритм враховує процеси дифузії по адатомним вузлам кристалічної решітки, та також ефект затінення осаджених атомів нерівностями рельєфу поверхні. Проведені розрахунки показують гарне згоду з наявними експериментальними результатами.

В работе разработана модель осаждения тонких плёнок, основанная на методе кинетического Монте-Карло. Предложенный алгоритм учитывает процессы диффузии по адатомным узлам кристаллической решетки, а также эффект затенения осаждаемых атомов неровностями рельефа поверхности. Проведенные расчеты показывают хорошее согласие с имеющимися экспериментальными результатами.

We developed a model of deposition of thin films based on the method of kinetic Monte Carlo. The proposed algorithm takes into account the processes of diffusion adatomnym lattice sites, as well as the effect of shadowing of the deposited atoms by the roughness of the surface relief. The simulations are in a good agreement with available experimental results.

**Введение.** Метод физического вакуумного осаждения (Physical Vapor Deposition, PVD) в настоящее время широко применяется в различных областях промышленности. PVD используется для получения многослойных материалов с хорошими характеристиками, которые применяются в производстве высокочувствительных магнитных слоев головок жестких дисков и магнитной памяти случайного доступа [1]. Данный метод получения пленок используется в производстве полупроводниковых устройств для получения многослойных устройств с субмикронными свойствами.

Производство тонких пленок при помощи физического вакуумного осаждения является процессом, который сильно зависит от условий роста пленок. Необходимость оптимизации процессов получения плёнок с заданными свойствами вызывает пристальное внимание к методам их компьютерного моделирования.

В настоящее время для моделирования роста пленок обычно широко используются методы молекулярной динамики баллистического осаждения [2-3]. Метод молекулярной динамики на текущий момент наиболее точен, однако он требует больших вычислительных мощностей и, как следствие, позволяет моделировать только небольшое количество осаждаемых атомов за малые интервалы времени. Метод баллистического

осаждения рассматривает только процессы осаждения атомов, не учитывая процессы диффузии, что не позволяет получить адекватные результаты для PVD. В кинетическом методе Монте-Карло рассматриваются как процессы осаждения, так и процессы диффузии, однако из-за особенностей построения алгоритма решения не требуется больших вычислительных ресурсов. Данная особенность позволяет рассматривать большое количество осадений за сравнительно небольшое время.

Поэтому целью данной работы была разработка компьютерной модели процессов атомарного осаждения тонких пленок кинетическим методом Монте-Карло (КМК).

**Физическая модель гомоэпитоксиального осаждения.** В данной работе рассматривались процессы осаждения тонких пленок на подложку из такого же материала, образующие кристаллическую структуру.

Рассматривается гомоэпитоксиальный рост тонких пленок. Так как из экспериментов [4] известно, что кристаллическая структура не изменяется с ростом пленки, поэтому в модели предполагается, что атомы могут занимать только идеальные кристаллографические позиции.

Кристаллическая решетка описывается позицией атома в элементарной ячейке и векторами трансляций.

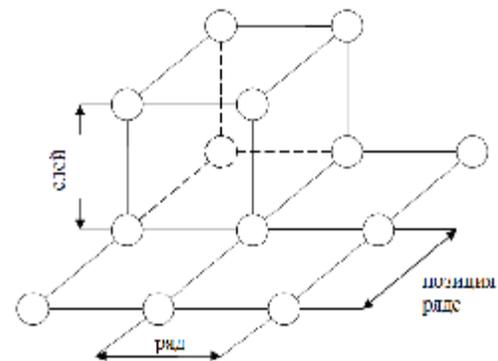


Рис 1. Схема построения кристаллической решетки

В общем случае ячейка представляет собой параллелепипед с векторами  $\mathbf{a}(a_x, a_y, a_z)$ ,  $\mathbf{b}(b_x, b_y, b_z)$ ,  $\mathbf{c}(c_x, c_y, c_z)$ . Для оптимальной работы алгоритма кристаллическая решетка стоилась заполнением по слоям и рядам (см. рис 1). Это позволяет однозначным образом определить соседние атомы по положению в плоскости, ряду и места в нем (т.е. отпадает необходимость перебора всех узлов для нахождения необходимого).

Каждый атом характеризуется энергией в узле  $e_s$ , которая вычисляется как сумма парных взаимодействий с первыми и вторыми соседями.

Процесс формирования поверхности может быть описан в рамках процессов осаждения атомов на поверхность и их поверхностной миграции.

Атомы осаждаются под некоторым углом  $q$ , рассчитываемым от нормали к поверхности осаждения. Осаждение происходит с постоянной скоростью  $F$ , и распределяются равномерно в плоскости, параллельной подложке.

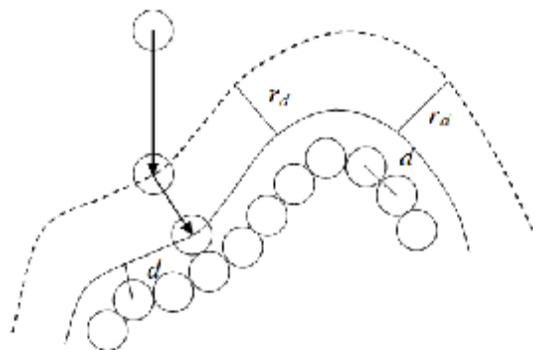


Рис 2. Схема процесса осаждения атомов.  $d$  – расстояние между атомами,  $r_d$  – расстояние, характеризующее взаимодействие между атомами

На рис. 2 приведена схема процесса осаждения. Считается, что атом движется по прямой до контакта с ближайшим поверхностным атомом. Критерием контакта является расстояние  $(d + r_d)$  между ними, где  $d$  – удвоенный атомный радиус, а величина  $r_d$  характеризует межатомное взаимодействие [5]. То, что  $r_d$  в модели отлично от нуля может приводить к морфологической нестабильности поверхности осаждаемых пленок. Обычно данный параметр в баллистическом осаждении не учитывается, что может приводить к неправильному описанию эволюции пленок. Если расстояние  $r < r_d$ , то атом занимает один из ближайших незанятых узлов. Конкретная позиция выбирается случайным образом с учетом энергии связи  $e_s$ .

В кинетическом методе Монте-Карло поверхностная диффузия является термически зависимым процессом, с вероятностью скачка из положения  $i$  в положение  $j$  описываемой Ареновской зависимостью [6]:

$$R^{ij} = R_0 \exp\left(\frac{E_A^{ij}}{k_B T}\right), \quad (1)$$

где  $R_0$  характеризует частоту атомных перескоков и составляет величину порядка  $10^{12} \text{c}^{-1}$ ,  $T$  – температура пленки и  $k_B$  – постоянная Больцмана,  $E_A^{ij}$  активационная энергия диффузионного скачка из положения  $i$  в положение  $j$ .

Активационная энергия  $E_A^{ij}$  была параметризована в терминах количества межатомных связей до и после диффузионного скачка следующим образом

$$E_A^{ij} = E_D + n_N E_N + n_{NN} + E_{NN}, \quad (2)$$

где  $E_D$  – диффузионный барьер скачка адатома,  $E_N$  и  $E_{NN}$  энергия связи первых и вторых соседей,  $n_N$  и  $n_{NN}$  – количество соседей до и после скачка.

**Алгоритм моделирования процессов формирования тонких пленок.** На основании вышеизложенного материала был разработан алгоритм моделирования процессов атомарного осаждения тонких пленок.

На рис. 3 приведены основные этапы моделирования процессов осаждения тонких пленок:

1. Задается структура кристаллической решетки, определяются геометрические координаты узлов, определяется начальное время осаждения  $t$  равное нулю.
2. В блоке 2 проводится проверка на синхронизацию процессов осаждения атомов и миграцию по времени  $t$ .
3. В третьем блоке случайным образом задаются начальные координаты атома в плоскости  $(x, y)$  находящейся на уровне нескольких слоев над последним заполненным слоем.
4. Последовательно изменяются координаты атома  $(x, y, z)$  с шагом  $\Delta X$ .
5. Проверяется, есть ли в ближайшем окружении другие атомы.
6. В блоке 6 случайным образом выбирается незанятая позиция в кристалле, которую занимает осаждаемый атом.
7. В массиве с состоянием узлов решетки позиция с данными координатами меняет свое значение на «занято»
8. Проверка времени  $t$  на критерий остановки процесса  $t_{cmon}$
9. В 9 блоке выбирается атом, который может мигрировать
10. Выбираются позиции, в которые атом может совершить скачек.
11. В блоке 11 выбирается случайным образом одна из позиций, полученных в блоке 10.
12. В массиве с состоянием узлов решетки выбранная позиция меняет свое значение на «занято», а предыдущая позиция принимает значение «свободно».

13. В блоке 13 добавляется к значению времени моделирования  $t$  значение времени миграции  $t_i$ .

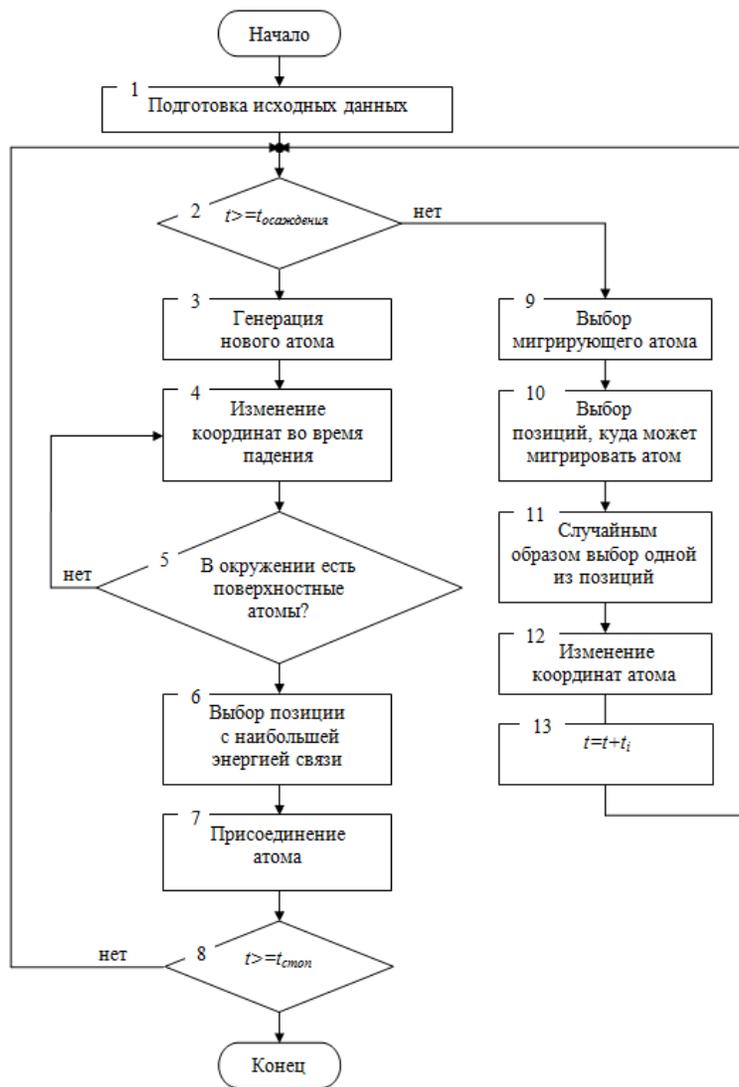


Рис 3. Блок-схема алгоритма моделирования процессов формирования тонких пленок

**Верификация алгоритма.** На основании описанного выше алгоритма было разработано программное средство, позволяющее моделировать рост поверхности тонких пленок. При помощи разработанного программного комплекса была проведена серия имитационных расчетов.

На рисунке 4 приведены результаты имитационного моделирования микрокристаллита, имеющего объемцентрированную кристаллическую решетку (ОЦК) после осаждения 1 млн. атомов. Как видно из рисунка, поверхность тонкой пленки имеет сложную морфологию, что обусловлено условиями процесса осаждения и микроструктуры пленки.

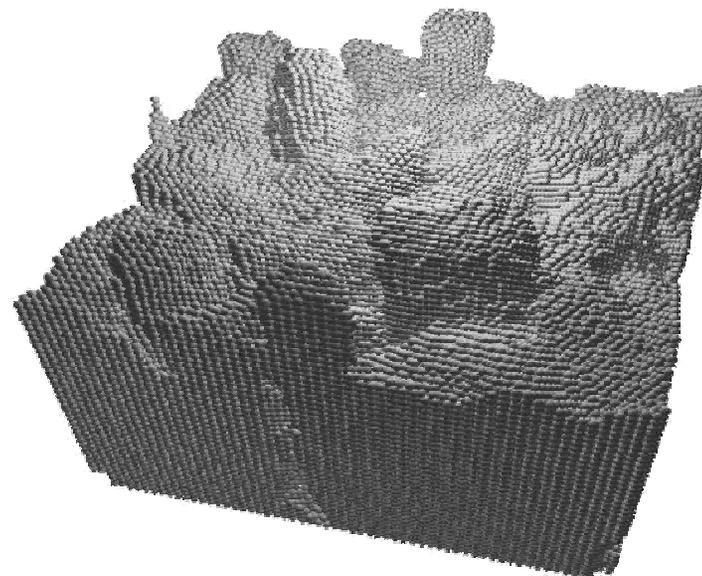


Рис 4. Результаты моделирования ОЦК-материала после осаждения 1 млн. атомов

Так же в рамках данной работы было проведено моделирование роста тонкой пленки с гранецентрированной кристаллической решеткой. На рисунке 5(а) приведены результаты компьютерного моделирования. На рисунке 5(б) приведены результаты эксперимента [4]. Как видно из рисунка, эти результаты находятся в качественном согласии с полученными результатами моделирования.

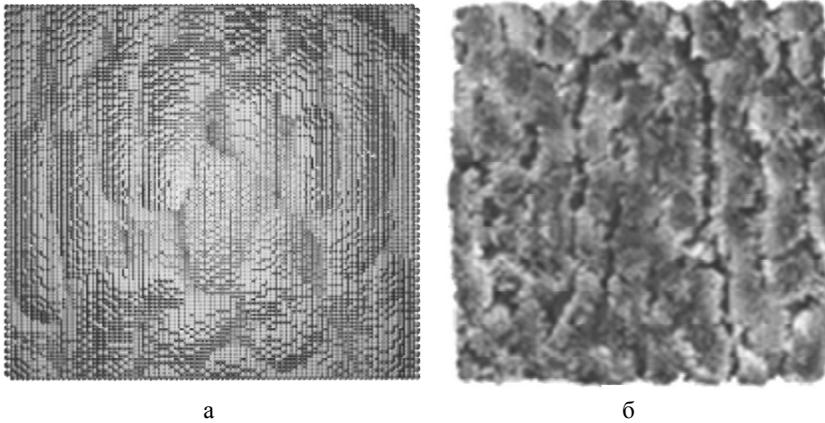


Рис 5. Сравнение результатов компьютерного моделирования (а) с экспериментальными данными (б)

**Выводы.** В настоящей работе разработана компьютерная модель формирования тонких пленок методом физического атомного осаждения. На основании созданной модели был создан алгоритм моделирования процессов атомарного осаждения и разработан программный комплекс.

При помощи созданного программного средства было проведено серия расчетов. Данные, полученные методами компьютерного моделирования, находятся в качественном согласии с имеющимися экспериментальными данными.

Данная разработка в дальнейшем может использоваться для изучения зависимости между условиями процесса осаждения и микроструктуры пленки.

**Список литературы:** 1. Mattox D.M., Handbook of Physical Vapor Deposition (PVD) Processing// William Andrew Publishing/Noyes -1998. -P.- 907. 2. Smith R. W., Srolovitz D. J. Void formation during film growth: A molecular dynamics simulation study// J. Appl. Phys.- 1996.-Vol. 79.- P. 1448-1457. 3. Srolovitz D. J., Mazor A., Bukiet B. G. Analytical and numerical modeling of columnar evolution in thin films //J. of Vac. Science & Technology.-1988.- Vol. A6.- P. 2371-2380. 4. Kohler U., Jensen C., Reshort K., et. al. Homo- and heteroepitaxy of metal on metal growth// Structure dynamics in heterogeneous systems.-Singapore: World Scientific, 2001.- P. 140-147. 5. Буддер К., Сиперли Д., Ансен Ж.-П. и др Методы Монте-Карло в статистической физике/ Пер. с англ.- М.: Мир, 1982.- 400 с. 6. Бокштейн Б.С., Бокштейн С.С., Жуховицкий А. А. Термодинамика и кинетика диффузии в твердых телах /М.: Металлургия, 1974.- 280 с.

Поступила в редакцию 18.12.09

УДК 621.3.088

**Н. Е. ХАЦЬКО**, аспирант НТУ «ХПИ»

## ОПТИМИЗАЦИЯ СТРУКТУРЫ МИНИМАЛЬНО-ИЗЫТОЧНОЙ СИСТЕМЫ НЕРАВНОТОЧНЫХ ИЗМЕРИТЕЛЕЙ

Отримано рішення задачі оптимального конструювання вимірювального модуля з мінімально надлишковою кількістю вимірів для оцінки вектора в умовах одиничних відмов датчиків, що розташовані на базових координатних осях.

Решена задача оптимального построения измерительного модуля с минимальной избыточностью измерений для оценки вектора при единичных отказах датчиков, расположенных на базовых координатных осях.

The problem of optimal design of measuring cell (or module) for estimating of measurable vector with minimal measuring redundancy was solved. The estimation was made under condition of single sensor failures.

**Введение.** В настоящее время в автоматических системах управления движением в качестве датчиков информации о движении все чаще используются инерциальные модули, выполненные на основе микроэлектромеханических систем (МЭМС) [1]. Такие устройства привлекают внимание широкого слоя потребителей доступностью и относительно невысокой стоимостью. Однако для таких применений, как космическая, ракетная, авиационная техника, точность подобных систем недостаточна. Для повышения точности измерений разрабатываются различные методы, основанные как на конструктивных, так и на алгоритмических решениях. Важным является также вопрос отказоустойчивости инерциальных модулей, вследствие чего в их состав включают дополнительные датчики. При этом, как правило, в состав модулей включают датчики одного класса точности [2]. В частности, в [3], [4] рассмотрены несколько вариантов построения избыточного по числу измерителей инерциального блока с использованием датчиков одинаковой точности.

Поскольку производство МЭМС-датчиков носит массовый характер, разброс точностных характеристик изделий одной и той же серии оказывается большим. Поэтому до использования измерителей в составе инерциального модуля следует провести уточнение их индивидуальных характеристик, после чего возникает задача выбора оптимальной структуры инерциального модуля в условиях неравноточных датчиков.

Следует отметить, что при использовании минимального числа измерителей – трех, задача оказывается тривиальной. Наилучшим ее решением является ортогональное расположение в модуле осей чувствительности датчиков. Задача становится значимой, если рассматривается избыточное число неравноточных измерителей.

Предлагается решение задачи оптимального конфигурирования минимально избыточной системы неравноточных измерителей по критерию минимальной погрешности измерений в случае отказа одного из датчиков. При этом предполагается, что вероятность отказа наиболее точного датчика пренебрежимо мала. С точки зрения практической реализации такое предположение допустимо, если в качестве наиболее точного датчика используется датчик другого класса, с заведомо более высокими характеристиками.

**Постановка задачи.** Рассматривается способ повышения точности измерений в условиях неравноточных датчиков на основе их избыточности.

В общем случае результатом работы инерциального модуля есть измерения вектора в трехмерном базовом пространстве, связанном с самим модулем. В составе модуля обычно имеются три измерителя, направление осей чувствительности которых в идеальном построении образуют ортогональный базис, и любой вектор однозначно определяется в этом пространстве его измеренными проекциями на оси чувствительности.

Зададим базовое ортогональное пространство и некоторый вектор в этом пространстве  $\mathbf{a} = \text{col}\{a_x, a_y, a_z\}$ . Вектор измерений обозначим  $\mathbf{a}^* = \{a_1^*, a_2^*, a_3^*\}$ .

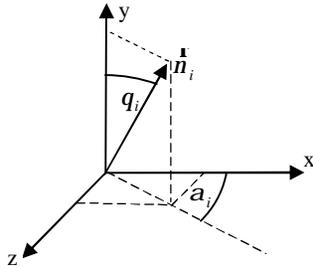


Рис. 1. Геометрическое расположение измерительных осей в пространстве

Минимально избыточными в трехмерном пространстве являются четыре измерения. В общем случае положение любой измерительной оси определено двумя углами  $a$  и  $q$ , что показано на рис. 1. Измерение каждой из них можно представить как разложение по компонентам в ортогональном базисе:  $\mathbf{n}_i = \{n_x^i, n_y^i, n_z^i\} = \{n_x^i \cdot n_1^i, n_y^i \cdot n_2^i, n_z^i \cdot n_3^i\}$ , где  $n_j^i$  – направляющие косинусы осей:  $n_1^i = \cos a_i \sin q_i$ ,  $n_2^i = \cos q_i$ ,  $n_3^i = \sin a_i \sin q_i$ ,  $i = \overline{1,4}$ .

Объединим измерения вектора  $\mathbf{a}$  всеми датчиками в вектор  $\mathbf{a}^* = \text{col}\{a_1^*, a_2^*, a_3^*, a_4^*\}$ , который связан с измеряемым вектором выражением

$$\mathbf{a}^* = G(\mathbf{a}, \mathbf{q}) \cdot \mathbf{a}, \quad (4)$$

где матрица  $G$  составлена из направляющих косинусов осей измерителей.

Комплексирование совокупности измерений проводится при помощи метода наименьших квадратов (МНК), тем самым получается оценка измеряемого вектора  $\hat{\mathbf{a}}$ :

$$\hat{\mathbf{a}} = (G^T G)^{-1} \cdot G^T \cdot \mathbf{a}^*. \quad (2)$$

В реальной системе измеренные значения вектора  $\mathbf{a}^*$  содержат в себе ошибки. Будем считать, что в модель ошибок аддитивно входят только случайные составляющие  $d\mathbf{a}^* = \{da_1^*, da_2^*, da_3^*, da_4^*\}$ , которые имеют нулевое математическое ожидание, дисперсии  $S_i^2, i = \overline{1,4}$  и не коррелированы между собой. Ошибка измерения вектора  $\hat{\mathbf{a}}$  в базисном пространстве будет составлять разность между комплексированным и истинным значением

$$d\hat{\mathbf{a}} = \hat{\mathbf{a}} - \mathbf{a}. \quad (3)$$

Учитывая аддитивный характер вхождения составляющих в математическую модель ошибки измерений, можно записать

$$\hat{\mathbf{a}} + d\hat{\mathbf{a}} = (G^T G)^{-1} \cdot G^T \cdot (\mathbf{a}^* + d\mathbf{a}^*),$$

тогда выражение для ошибок аналогично (2):

$$d\hat{\mathbf{a}} = (G^T G)^{-1} \cdot G^T \cdot d\mathbf{a}^*. \quad (4)$$

Ортогональное расположение осей трех датчиков является наилучшим для измерения вектора, поэтому оси чувствительности менее точных измерителей направим вдоль базовых осей. В этом случае матрица  $G$  имеет вид

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ или } G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ n_1 & n_2 & n_3 \end{bmatrix}, \quad (5)$$

где углы  $a$  и  $q$  определяют направление оси четвертого датчика и являются искомыми величинами.

Оценка измеряемого вектора  $\hat{\mathbf{a}}$  зависит от матрицы  $K(a, q) = (G^T G)^{-1} \cdot G^T$ , содержащей значения направляющих косинусов для четвертого измерителя. Введем скалярную меру точности оценки:

$$F(d\hat{\mathbf{a}}) = (d\hat{\mathbf{a}}, d\hat{\mathbf{a}}) = (K \cdot d\mathbf{a}^*, K \cdot d\mathbf{a}^*) \quad (6)$$

Очевидно, что наилучшая оценка измеряемого вектора в смысле минимума  $F(\hat{d}^{\mathbf{a}})$  получается при замене наименее точного измерителя на наиболее точный, при этом измерения наименее точного датчика игнорируются. Однако при возможном отказе одного из «грубых» датчиков такое решение приводит к потере полноты измерений трехмерного вектора.

В этих условиях формулируется задача определения такого положения оси четвертого датчика, чтобы в случае отказа одного из трех «грубых» датчиков точность оценка измеряемого вектора по показаниям оставшихся трех датчиков была одинаковой, не зависимо от номера отказавшего датчика. Иными словами, **решается задача поиска положения четвертой оси чувствительности**, при котором оценки измеряемого вектора при любом отказе одного из датчиков, расположенных на базовых координатных осях, будут осуществляться с одинаковой точностью.

**Анализ задачи.** Рассмотрим случаи отказа первого, второго или третьего датчика, на что в дальнейшем изложении будут указывать индексы. Такие отказы меняют вектор измерений  $\mathbf{r}^{\mathbf{a}}$  и матрицу (5): соответствующие строки матрицы  $G$  и вектора  $d\mathbf{a}^{\mathbf{a}}$  вычеркиваются и модель (4), (6) упрощается до вида

$$d\hat{\mathbf{a}}^{\mathbf{a}} = G_i^{-1} \cdot d\mathbf{r}_i^{\mathbf{a}}, \quad (7)$$

$$F(d\hat{\mathbf{a}}^{\mathbf{a}}) = \left( d\hat{\mathbf{a}}^{\mathbf{a}}, d\hat{\mathbf{a}}^{\mathbf{a}} \right) = \left( G_i^{-1} \cdot d\mathbf{r}_i^{\mathbf{a}}, G_i^{-1} \cdot d\mathbf{r}_i^{\mathbf{a}} \right), \quad i = 1, 2, 3. \quad (8)$$

Случайный характер ошибки измерений позволяет перейти к рассмотрению дисперсий оценок. Дисперсия ошибки оценки вектора при использовании четырех измерений имеет вид

$$D = s_1^2 \left( 1 - \frac{3}{4} n_1^2 \right) + s_2^2 \left( 1 - \frac{3}{4} n_2^2 \right) + s_3^2 \left( 1 - \frac{3}{4} n_3^2 \right) + \frac{1}{4} s_4^2. \quad (9)$$

Из (4) видно, что дисперсия ошибки для оценки, построенной по четырем измерениям, меньше, чем ошибка оценки по трем «грубым» измерителям, соответствующая выражению  $s_1^2 + s_2^2 + s_3^2$ . Отношение оценок выражается следующей зависимостью

$$\frac{D}{s_1^2 + s_2^2 + s_3^2} = 1 - \frac{1}{s_1^2 + s_2^2 + s_3^2} \cdot \left( \frac{3}{4} (s_1^2 n_1^2 + s_2^2 n_2^2 + s_3^2 n_3^2) + \frac{1}{4} s_4^2 \right) < 1. \quad (10)$$

Следовательно, введение в систему измерений четвертого более точного датчика целесообразно.

При отказе одного из трех датчиков выражение для дисперсии принимает вид:

$$\begin{aligned} D_1 &= \frac{s_2^2(1-n_3^2) + s_3^2(1-n_2^2) + s_4^2}{n_1^2} \\ D_2 &= \frac{s_1^2(1-n_3^2) + s_3^2(1-n_1^2) + s_4^2}{n_2^2}, \\ D_3 &= \frac{s_1^2(1-n_2^2) + s_2^2(1-n_1^2) + s_4^2}{n_3^2} \end{aligned} \quad (11)$$

В соответствии с введенным критерием в основу выбора расположения четвертого датчика полагаются условия

$$D_1 = D_2 = D_3. \quad (12)$$

Для достижения полноты системы уравнений воспользуемся связью направляющих косинусов для искомой оси, и запишем следующую систему

$$\begin{cases} D_2 - D_3 = 0 \\ D_1 - D_3 = 0 \\ n_1^2 + n_2^2 + n_3^2 = 1 \end{cases}. \quad (13)$$

**Решение системы.** Введем новые переменные:  $x = \frac{n_1^2}{n_2^2}$ ,  $y = \frac{n_2^2}{n_3^2}$ ,  $z = \frac{1}{n_3^2}$ .

В результате система уравнений (13) относительно трех переменных  $x, y, z$  принимает вид:

$$\begin{cases} x(s_1^2 + s_4^2) + y(s_2^2 + s_4^2) - \frac{1}{x}(s_3^2 + s_4^2) - \frac{y}{x}(s_2^2 + s_4^2) + s_1^2 - s_3^2 = 0 \\ \frac{1}{y}(s_3^2 + s_4^2) + \frac{x}{y}(s_1^2 + s_4^2) - \frac{1}{x}(s_3^2 + s_4^2) - \frac{y}{x}(s_2^2 + s_4^2) + s_1^2 - s_2^2 = 0 \\ x + y + 1 = z \end{cases} \quad (14)$$

Последовательно исключая переменные, получим кубическое уравнение

$$\begin{aligned} &x^3(s_1^2 - s_2^2)(s_1^2 + s_4^2) + x^2(s_1^2 + s_4^2)(s_1^2 + s_2^2 - 2s_3^2) + \\ &+ x(s_3^2 + s_4^2)(s_2^2 + s_3^2 - 2s_1^2) + (s_3^2 + s_4^2)(s_3^2 + s_2^2) = 0 \end{aligned} \quad (15)$$

После приведения его к нормальному виду и использования подстановки  $x = 1 - \frac{s_1^2 + s_2^2 - 2s_3^2}{3(s_1^2 - s_2^2)}$  уравнение (15) решается методом Кардано.

Численные примеры решения задачи приведены в таблице 1.

Таблица 2

**Моделирование.** Приведем пример применения изложенного подхода при оценивании величины вектора  $\vec{a}$  по четырем неравноточным измерениям. Излагаемые ниже результаты получены численным моделированием. Динамика во времени измеряемого вектор  $\vec{a}$  задается своими проекциями, меняющимися по закону

$$a_x^* = A + da_x(t)$$

$$a_y^* = B + da_y(t),$$

$$a_z^* = C + da_z(t)$$

где  $A, B, C$  – константы. Данный вектор измеряется четырьмя датчиками, и их случайные погрешности определяются дисперсиями, указанными в четырех первых столбцах таблицы 1. Три первых датчика расположены вдоль базисных осей, расположение четвертого найдено из условия равноточности оценки при отказе одного из «грубых» датчиков. Таким образом, в модели заданы найденные при решении системы уравнений (13) оптимальные углы наклона оси четвертого измерителя. Эти углы, рассчитанные для различных сочетаний точностей базисных измерителей, приведены в таблице 1.

значение	дисперсии оценок по измерениям:				
	трем	четырем	отказ 1 <sup>го</sup> дат.	отказ 2 <sup>го</sup> дат.	отказ 3 <sup>го</sup> дат.
теоретическое	0,84	0,739	1,0	1,0	1,0
1-я реализация	0,85	0,740	1,02	1,02	1,02
2-я реализация	0,85	0,742	0,96	1,02	1,02

Таблица 1

Точность датчиков, дисперсии осей				значение угла $a$ , град.	значение угла $q$ , град.
$s_1^2$	$s_2^2$	$s_3^2$	$s_4^2$		
0,64	0,04	0,16	0,01	59,87	39,48
0,64	0,16	0,04	0,01	67,54	56,64
0,04	0,64	0,16	0,01	35,47	71,39
0,3	0,36	0,2	0,01	48,91	58,23
0,36	0,3	0,2	0,01	50,59	56,03
0,36	0,2	0,3	0,01	46,71	50,13
0,25	0,25	0,25	0,01	45,0	54,74

В каждой точке для оценки величины измеряемого вектора  $\vec{a}$  измерения обрабатывались с помощью МНК. В результате моделирования получены 5 рядов, соответствующих оценкам величины вектора  $\vec{a}$  в течении времени при различных сочетаниях датчиков. Как показано в таблице 2 и на рис. 2, апостериорные значения дисперсий измерений при отказах датчиков сравнимы между собой, и их можно округлить до априорного теоретического значения. Дисперсии оценок по базисным и избыточным измерениям близки к своим теоретическим значениям, рассчитанным по формулам (9), (11).

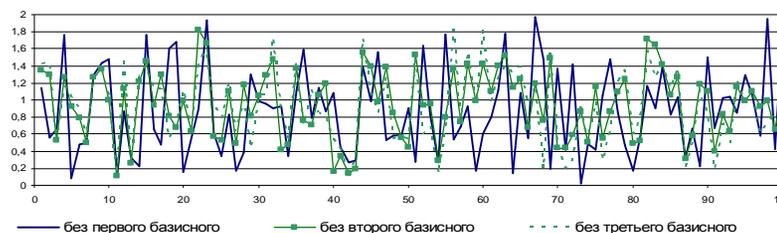


Рис. 2. Погрешности оценок измерений в случае отказа одного из датчиков, расположенных на базисных осях.

**Выводы.** Таким образом, после применения алгоритма комплексирования оценка измеряемого вектора улучшается, степень повышения точности зависит от точности дополнительно введенного измерителя. Найдена зависимость положения четвертой оси чувствительности для равноточной оценки измеряемого вектора при любом отказе одного из датчиков, расположенных на базисных координатных осях. Результаты моделирования подтверждают теоретические выводы.

**Список литературы:** 1. Кробка Н. И., Мезенцев А. П., Мезенцев О. А. Методики калибровки микроэлектромеханических инерциальных измерительных блоков в массовом производстве. // XIV С-Пб. междунар. конф. по интегрированным навигационным системам. – СПб.: ГНЦ РФ ЦНИИ «Электроприбор», 28-30 мая 2007. С. 96-99. 2. Бранец В. Н., Дибров Д. Н. и др. Измеритель вектора кажущегося линейного ускорения – прибор БИЛУ КХ69-042 для СУ спускаемого аппарата «Союз-ТМА» // XIII С-Пб. Междунар. конф. по интегрированным навигационным системам. – СПб.: ГНЦ РФ ЦНИИ «Электроприбор». – 2006. – С. 253-263. 3. Мумин О. Л., Дроздов М. В., Рябова Л. П. и др. Трехкоординатный малогабаритный акселерометр с цифровым выходом повышенной надежности // XVI С-Пб. междунар. конф. по интегрированным навигационным системам. – СПб.: ГНЦ РФ ЦНИИ «Электроприбор», – 2009. – С. 66-68. 4. Бельский Л. Н., Водичева Л. В. Повышение точности гироскопирования за счет выбора ориентации осей чувствительности измерителей // Гироскопия и навигация. – 2000. -№ 3. – С. 3-16.

Поступила в редакцию 30.11.09

**В. Б. УСПЕНСКИЙ**, канд. техн. наук,  
**А. Д. АСЮТИН**, аспирант НТУ «ХПИ»

### РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ДВИЖЕНИЯ ИНЕРЦИАЛЬНОГО БЛОКА НА ДВУХОСНОМ ПОВОРОТНОМ СТОЛЕ

Розроблено імітаційну модель руху ІБ, встановленого на двовісний поворотний стіл (ДПС). Модель дозволяє генерувати показники інерціальних датчиків, адекватні їх реальній роботі, при різних видах руху, характерних для ДПС. Дані, що генеруються моделлю, можуть використовуватися для створення і відладки алгоритмів паспортизації ІБ.

Разработана имитационная модель движения ИБ, установленного на двухосный поворотный стол (ДПС). Модель позволяет генерировать показания инерциальных датчиков, адекватные их реальной работе, при различных видах движения, характерных для ДПС. Генерируемые моделью данные могут использоваться для создания и отладки алгоритмов паспортизации ИБ.

The simulation model of the motion of IB, set on a biaxial turning table (BTT), is developed. The model allows to generate the testimonies of inertial sensors, adequate to their real work, sight different, characteristic for BTS motion. The information generated by a model can be used for creation and debugging of algorithms of the passport system of IB.

**Введение.** В состав бесплатформенной инерциальной навигационной системы, как правило, входят три гироскопа и три акселерометра, с помощью которых соответственно измеряется вектор угловой скорости вращения и вектор кажущегося ускорения [1]. Указанные измерения реализуются с ошибкой, обусловленной не идеальностью самого датчика, его электронных компонент, преобразующих сигнал, а также погрешностью установки датчика в инерциальном блоке (ИБ).

Для разделения совокупной ошибки измерения по перечисленным факторам используется *система инструментальных погрешностей* инерциального блока, включающая в себя:

- *смещение «нуля» датчиков*, под которым понимается величина выходного сигнала при отсутствии измеряемого фактора;
- *погрешность масштабного коэффициента* преобразования аналогового выходного сигнала датчика в цифровой код;
- *угловые параметры несоосности осей чувствительности датчиков* с осями виртуальной ортогональной системы координат, принимаемой в качестве базовой.

Изменение во времени перечисленных погрешностей носит сложный характер. В их динамике можно выделить «быструю» и «медленную» составляющие. «Быстрая» составляющая зависит от внешних факторов, таких как температура и скорость ее изменения, магнитное поле, линейное и угловое ускорение и др. «Медленная» связана с выработыванием ресурса датчика и

обуславливается изменением в ходе эксплуатации свойств материала, из которого он изготовлен, используемых в нем сред, износом элементов.

Для повышения точности изменений необходимо оценивать указанные погрешности и алгоритмически их компенсировать. Такая оценка осуществляется, как правило, в три этапа.

Первый этап реализуется после изготовления инерциального блока в заводских (лабораторных) условиях. В ходе этого этапа оценивается «быстрая» составляющая погрешностей, выявляется и параметризуется ее зависимость от внешних факторов, критичных для данного датчика. Такую формальную зависимость будем называть *математической моделью* погрешностей, а процедуру определения ее параметров - *паспортизацией* блока. Процедура паспортизации характерна большим объемом измерений, она занимает продолжительное время и требует специального оборудования и метрологического обеспечения. В состав используемого при этом высокоточного оборудования входят, как правило, одно-, двух- или трехосные поворотные стенды. Второй этап – регламентный контроль и корректировка параметров модели в ходе эксплуатации. Третий этап – оценка текущих значений остаточных погрешностей, некомпенсированных с помощью математической модели, непосредственно в процессе функционирования. Проведение и обработка измерений на каждом этапе производится по своим методикам [2,3]. Несмотря на многочисленные примеры, описанные в литературе, разработка эффективной методики паспортизации и докалибровки ИБ навигационных систем, особенно с использованием относительно недорогого лабораторного оборудования, остается актуальной.

**Постановка задачи.** Данная статья связана с разработкой методического и алгоритмического обеспечения паспортизации ИБ с использованием специального оборудования – двухосного поворотного стола (ДПС) и посвящена разработке имитационной модели движения ИБ, установленного на ДПС. Такая модель, реализованная программно, позволяет генерировать показания инерциальных датчиков, адекватные их реальной работе, при различных видах движения, характерных для ДПС. Генерируемые моделью данные необходимы для создания и отладки алгоритмов паспортизации ИБ. Разработанное программное средство позволяет также выявить последовательность тестовых движений ДПС для наиболее эффективной паспортизации блока.

Таким образом, решаемая в данной статье задача непосредственно связана с повышением точности функционирования инерциальных навигационных систем.

**Описание алгоритма.** Общая методика построения имитационной модели, использованной далее, изложена в [4]. В данной статье рассматривается частный случай движения, обеспечиваемого ДПС, упрощенная схема которого приведена на рис.1, для которого доработана упомянутая модель.

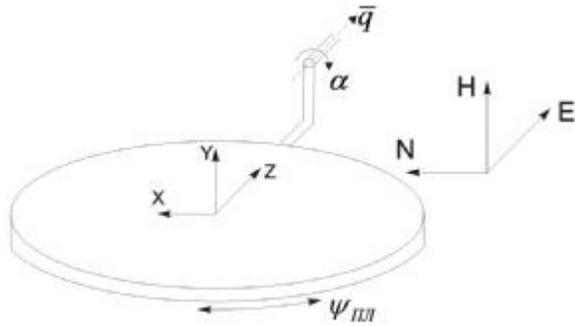


Рис. 1. Схема ДПС

Поворотная платформа ДПС, на которую устанавливается ИБ, имеет две степени свободы: поворот вокруг оси Y на угол  $\psi_{\text{пл}}$  и поворот платформы вокруг оси, совпадающей с вектором  $\vec{q}$ , жестко связанным с неподвижной частью ДПС, на угол  $a$ . Данная установка имеет ряд особенностей и ограничений на движение, а именно:

- осуществлять вращение вокруг оси Y возможно только в случае, когда угол  $a$  равен нулю;
- угол  $a$  может меняться в пределах  $[-2p; 2p]$ ;
- точно известна начальная азимутальная ориентация ИБ.

Отметим, что с помощью данного поворотного стола можно реализовать любую ориентацию ИБ в пространстве.

Поворачивая платформу на углы  $\psi_{\text{пл}}$  и  $a$ , необходимо определить углы курса, тангажа и крена ИБ, установленного на платформе, а также показания его датчиков в процессе угловых маневров. При решении данной задачи будем предполагать, что географические координаты платформы (долгота и широта) остаются неизменными.

Чтобы обеспечить неразрывность и целостность выходных данных имитатора движения, введем зависимости  $\psi_{\text{пл}}(t)$  и  $a(t)$  в виде полиномов третьего и пятого порядка соответственно.

$$\psi_{\text{пл}}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3,$$

$$a(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5;$$

Функция  $\psi_{\text{пл}}(t)$  должна быть неразрывной вместе со своей первой производной. Таким образом, параметры  $\psi_{\text{пл}}(t)$  будем определять, исходя из краевых условий для самой функции и ее первой производной.

Соответственно параметры  $a(t)$  будем определять, исходя из краевых условий для самой функции, ее первой и второй производной.

На рисунке 2 приведена блок-схема алгоритма определения выходных данных модели ДПС.

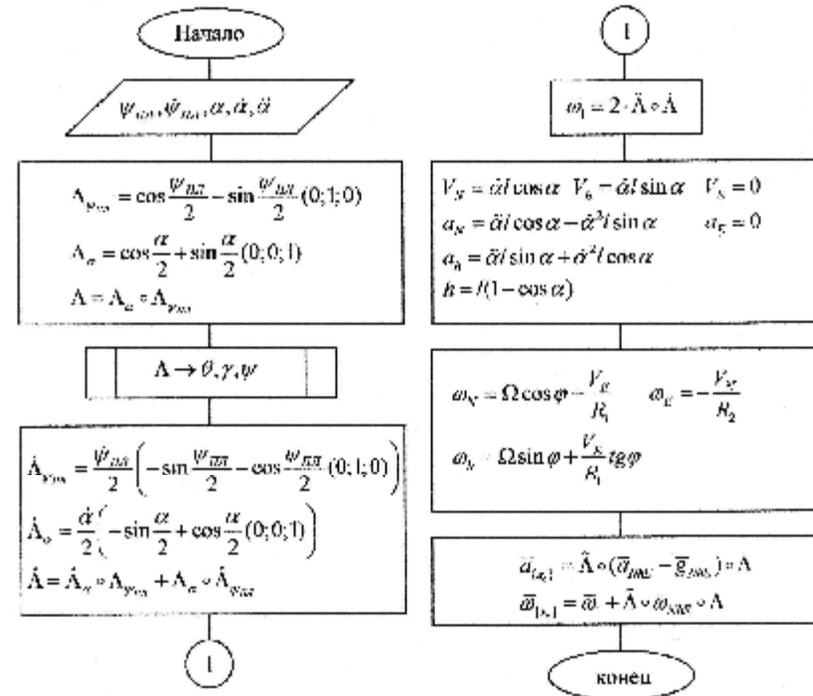


Рис. 2. Блок-схема алгоритма определения НП и показаний ИБ

В алгоритме использованы следующие обозначения:

$\psi_{\text{пл}}, \psi_{\text{пл}}', a, \alpha, \alpha'$  – переменные, отвечающие текущему состоянию ДПС,

$\Lambda_{\psi_{\text{пл}}}, \Lambda_{\psi_{\text{пл}}}', \Lambda_a, \Lambda_a'$  – вспомогательные переменные,

$\Lambda, \Lambda'$  – кватернион ориентации ИБ относительно местной географической системы координат и его производная,

$q, g, \gamma$  – углы тангажа, крена и курса соответственно,

$\omega_i$  – угловая скорость вращения ИБ в местной географической системе координат,

$V_N, V_h, V_E, a_N, a_h, a_E$  – составляющие скорости и ускорения ИБ в местной географической системе координат,

$l$  – длина ручки ДПС,

$h$  – высота места расположения ИБ,

$w_N, w_h, w_E$  – составляющие вектора угловой скорости вращения местной географической системы координат,

$\vec{g}_{NhE}$  – вектор ускорения свободного падения

$a_{\{x_c\}}$  – вектор кажущегося ускорения ИБ в проекциях на оси чувствительности ИБ,

$w_{\{x_c\}}$  – вектор угловой скорости вращения ИБ в проекциях на оси чувствительности ИБ.

В данном алгоритме параметры  $\kappa, R_1, R_2$  рассчитываются по следующим формулам:

$$k = \sqrt{1 - e^2 \sin^2 j}, \quad R_1 = \frac{a_0 + h}{k}, \quad R_2 = \frac{(a_0 + h)(1 - e^2)}{k^3}$$

где  $a_0 = 6378137\text{м}$  – большая полуось земного эллипсоида;

$e^2 = 0.0066943800$  – квадрат первого эксцентриситета земного эллипсоида;

$j$  – географическая широта места.

Вектор  $\vec{g}_{NhE}$  считаем постоянным в силу неизменности географической широты места расположения ИБ.

В алгоритме используется процедура получения углов курса, тангажа и крена по компонентам кватерниона ориентации [5]:

$$q = \arcsin(2I_0I_3 + 2I_2I_1),$$

$$\gamma = \text{sign}(r_1) \cdot \arcsin \frac{-2I_0I_2 + 2I_3I_1}{\cos q} + \frac{p}{2}(1 - \text{sign}(r_1)),$$

$$g = \text{sign}(r_2) \cdot \arcsin \frac{2I_0I_1 - 2I_3I_2}{\cos q} + \frac{p}{2}(1 - \text{sign}(r_2)),$$

где  $r_1 = I_0^2 + I_1^2 - I_2^2 - I_3^2$ ,  $r_2 = I_0^2 + I_2^2 - I_1^2 - I_3^2$ ,  $I_j, j = \overline{0,3}$  – компоненты кватерниона  $\Lambda$ .

**Результаты моделирования.** Рассмотрим типичные результаты работы имитационной модели. С помощью алгоритма промоделировано некоторое простое движение, для которого известно изменение навигационных параметров (НП). Пусть платформа в течение 5 секунд поворачивается вокруг

оси Y на  $90^\circ$ , затем 1 секунду находится в состоянии покоя, после чего поворачивается в течение 4 секунд вокруг оси  $\vec{q}$  на  $90^\circ$ . Такому движению соответствует изменение угла курса  $\gamma$  от  $0^\circ$  до  $90^\circ$  в интервале времени от 0с до 5с, а также изменение угла крена  $g$  от  $0^\circ$  до  $90^\circ$  в интервале времени от 6с до 10с. При вращении платформы вокруг оси  $\vec{q}$  также будет изменяться высота  $h$ . В данной реализации длина «ручки» ДПС принималась равной 1м, что приводит к изменению высоты на 1м. В результате моделирования имитатор генерирует показания ИБ, как на рис.3.

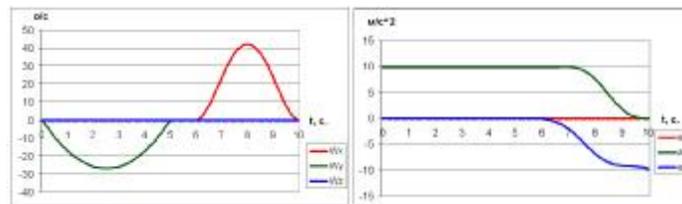


Рис. 3. Сгенерированные имитатором движения показания ИБ

Для проверки правильности функционирования имитатора движения ДПС используем сгенерированные значения в качестве исходных данных для алгоритмов бесплатформенной инерциальной навигационной системы (БИНС). Эти алгоритмы, по сути, решают «обратную» задачу, а именно, интегрируя уравнения движения, формируют навигационные параметры, приведенные на рис.4.

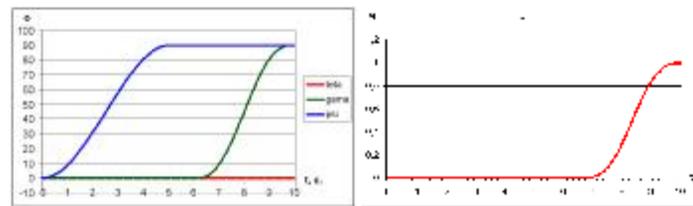


Рис.4. Навигационные параметры, полученные в БИНС

Из результатов следует, что навигационные параметры, вычисляемые БИНС по сгенерированным моделью измерениям ИБ, совпадают с их эталонными значениями. Имитационная модель тестировалась на различных видах движения, сгенерированные ею данные были проверены на адекватность.

**Выводы.** Созданная имитационная модель движения ДПС отвечает следующим требованиям:

- генерирует показания инерциальных датчиков, адекватные их реальной работе;
- генерирует точные (эталонные) значения навигационных параметров ИБ;
- позволяет моделировать различные виды движения, характерные для данной установки.

Выходные данные имитационной модели: угловая скорость и кажущееся ускорение в проекциях на приборные оси ИБ, будут использованы для создания и отладки алгоритмов паспортизации ИБ.

**Список литературы:** 1. Ануцин О.Н., Емельянец Г.И. Интегрированные системы ориентации и навигации для морских подвижных объектов. Санкт-Петербург 1999. – 357с. 2. Анисейчев В.С., Атаманов Н.А., Поликовский Е.Ф. Методика калибровки акселерометров. Сборник статей XXII научно-технической конференции памяти Н.Н. Острякова. 3. Крузе С.В., Лене С.Н., Молчанов А.В., Поликовский Е.Ф. Калибровка триады лазерных гироскопов. "Гироскопия и навигация" №4 (55), 2006. – 87с. 4. Брославец Ю.Ю., Фомичев А.А., Колчев А.Б., Успенский В.Б. Имитационная модель движения летательного аппарата. Сборник статей IX Санкт-Петербургской международной конференции по интегрированным навигационным системам. 5. Успенский В.Б. Теоретические основы гиросилового управления ориентацией космического летательного аппарата. Монография. – Харьков: НТУ «ХПИ». - 2006. – 328с.

*Поступила в редакцию 18.09.09*

УДК 621.384:66.045.1:536.27

**А. М. ГАНЖА**, канд. техн. наук,  
**Н. А. МАРЧЕНКО**, канд. техн. наук

### **КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ У СКЛАДНИХ ТЕПЛООБМІННИХ АПАРАТАХ**

В статті пропонуються математична модель та алгоритми для комп'ютерного моделювання складних теплообмінних апаратів. Результати моделювання зіставлені з відомими даними та можуть використовуватись для системного аналізу процесів у теплообмінних апаратах.

В статье предлагаются математическая модель и алгоритмы для компьютерного моделирования сложных теплообменных аппаратов. Результаты моделирования сопоставлены с известными данными и могут использоваться для системного анализа процессов в теплообменных аппаратах.

In the article there have been proposed mathematical model and algorithms for simulation of difficult heat-exchange apparatus. The simulation results are compared with known data and may be used for system analysis of process in heat-exchange apparatus.

**Вступ.** У теперішній час теплообмінні апарати широко використовуються у різних галузях національної економіки. Для аналізу теплового стану

теплообмінників необхідно знати локальні температурні характеристики у різних точках апарату. Це дозволить аналізувати їх ефективність на різних режимах експлуатації, а також прогнозувати їх ресурс з урахуванням експлуатаційних факторів, якими можуть виступати забруднення, відкладення, корозійно-ерозійний знос, контактні термічні опори та ін. Необхідність отримання локальних температурних характеристик вимагає використання методів комп'ютерного моделювання.

Теплообмінне обладнання належить до класу складних технічних систем, тому що може компонуватися з багатьох секцій, кожна з яких може бути класифікована за плинном теплоносіїв, що гріє та нагрівається. Для наближення схеми до найбільш ефективної, тобто протитокової, у теплообмінниках з перехресним плинном використовуються багатоходові схеми компонування. При достатній кількості ходів взаємний рух теплоносіїв наближається до класичного протитоку. Така схема взаємного плинну теплоносіїв називається змішаною.

Для теплообмінників з простою змішаною схемою, де кожен хід представляє собою будь-які однакові теплообмінники (секції), залежності для побудови розподілу температур, визначення ефективності апарату описані у роботах [1, 2 та ін.]. У цій схемі припускається, що обидва теплоносія між ходами цілком перемішуються.

На практиці найбільш розповсюдженою схемою компонування теплообмінників є складна змішана схема току. На відміну від простої змішаної схеми тут один з теплоносіїв рухається окремими струменями за всією довжиною та між ходами не перемішується, а теплоносіїв, що знаходиться усередині, у межах одного ходу як правило рухається окремими струменями та між ходами цілком перемішується. Таким чином, схема змішаного плинну буде складною, де кожен хід (секція) представляє собою складну схему однократного перехресного плинну. Для таких схем залежності для побудови розподілу температур на вході та виході з кожного ходу, визначення ефективності апарату у відомих роботах не приводяться (крім частинних випадків у [2]). Тому математичне моделювання температурних характеристик та ефективності таких апаратів є актуальною задачею.

**Постановка задачі.** У даній роботі поставлена задача розробки математичної моделі для складних теплообмінних апаратів та створення методики, що дозволяє одержати розподіл температурних характеристик по кожному окремому елементу теплообмінника і в апараті в цілому. Створена методика повинна дати можливість проводити системний аналіз, як знову проєктованих, так і діючих теплообмінників, а також здійснювати оптимізацію характеристик апаратів.

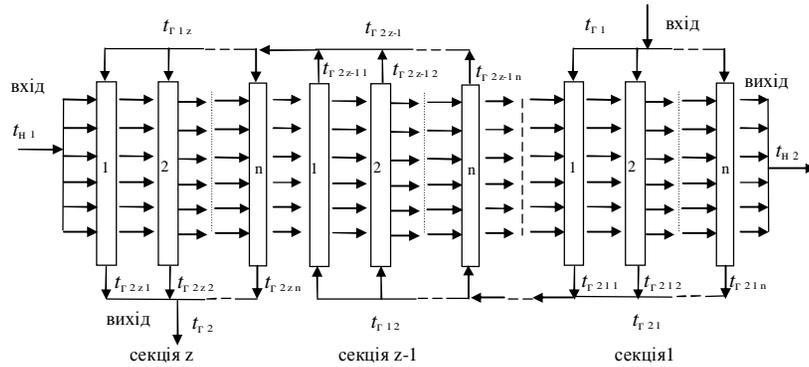
**Розв'язання проблеми.** Апарати, що розглядаються, можуть мати різне компонування теплообмінних поверхонь: рівнобіжне чи послідовне протитокове включення секцій по зовнішньому теплоносію, причому секції можуть бути одноходовими чи багатоходовими. У межах однієї секції

зовнішній теплоносій цілком практично не переміщується. Ступінь неперемішування зовнішнього теплоносія визначається кількістю ребер, що приходяться на одиницю довжини ряду труб. У межах однієї секції внутрішній теплоносій переміщується частково – тільки в межах ряду труб одного ходу. Між ходами та секціями внутрішній теплоносій, як правило, переміщується.

Для теплообмінника, де коефіцієнти теплопередачі однакові в межах одного ходу в секції при нескінченному числі струменів зовнішнього теплоносія, авторами отримано аналітичний розв'язок для розподілу локальних температур і на їх основі розроблено алгоритми визначення ефективності. Вихідна система рівнянь включає рівняння теплового балансу і теплопередачі для елементарних ділянок (елементів) поверхні, а також інтегральні рівняння, що враховують особливості руху теплоносіїв.

Розглянемо найбільш складний випадок компоновання теплообмінника, (див. рис. 1). Такі теплообмінники є дуже розповсюдженими (одноходові багатосекційні калорифери, апарати повітряного охолодження та ін.).

У розробленому авторами аналітичному алгоритмі визначаються локальні температури середовища, що гріє та ефективність всього теплообмінника:



$t$  – температура; "Г" – теплоносій, що гріє; "н" – теплоносій, що нагрівається

Рисунок 1 – Одноходова багатосекційна схема з нескінченним числом струменів

а) для першого ряду в секції ( $i = 1$ ):

$$\begin{aligned} \overline{t_{Г\bar{x}1k}} &= \frac{t_{Г1k} - t_{Г\bar{x}1k}}{t_{Г1} - t_{н1}} = \\ &= e^{-R \cdot a \cdot \bar{x}} \cdot \omega_{1k} = (1 - e^{-R \cdot a \cdot \bar{x}}) \cdot (1 - \overline{\varphi}_{(k-1)}) + \theta_{\text{співп}} + \theta_{\text{прот}}, \end{aligned} \quad (1)$$

$$\theta_{\text{співп}} = -\left(\frac{a}{n}\right) \cdot \sum_l \sum_{i=1}^n \left( \overline{t_{Г\bar{x}il}} - Ra \cdot e^{-R \cdot a \cdot \bar{x}} \cdot \int_0^{\bar{x}} e^{R \cdot a \cdot \bar{x}} \overline{t_{Г\bar{x}il}} d\bar{x} \right), \quad (2)$$

$$\theta_{\text{прот}} = \left(\frac{a}{n}\right) \cdot \sum_m \sum_{i=1}^n \left( \overline{t_{Г(l-\bar{x})im}} - e^{-R \cdot a \cdot \bar{x}} \cdot \overline{t_{Г(\bar{x}=1)im}} - Ra \cdot e^{-R \cdot a \cdot \bar{x}} \cdot \int_0^{\bar{x}} e^{R \cdot a \cdot \bar{x}} \overline{t_{Г(l-\bar{x})im}} d\bar{x} \right), \quad (3)$$

де  $l$  – номери співпадаючих по напрямку з координатою  $x$  від входу в елемент усіх попередніх секцій;  $l = z, z-2, z-4 \mathbf{K}(k-2)$ , якщо напрямок руху внутрішнього теплоносія збігається з напрямком в останній секції  $z$ ;  $i = l = z-1, z-3 \mathbf{K}(k-2)$ , у протилежному випадку, тобто якщо напрямок руху внутрішнього теплоносія в поточній секції не збігається з напрямком в останній секції  $z$ ;

$m$  – номери протилежних за напрямком з координатою  $x$  від входу в елемент усіх попередніх секцій;  $m = z-1, z-3 \mathbf{K}(k-1)$ , якщо напрямок руху внутрішнього теплоносія збігається з напрямком в останній секції  $z$ , і  $m = z, z-2, z-4 \mathbf{K}(k-1)$ , у протилежному випадку.

$\bar{x}$  – відносна координата від входу внутрішнього середовища в межах секції,  $\bar{x} = \frac{x}{L}$ ,  $L$  – довжина ряду труб у секції;  $i$  – номер ряду в секції,  $n$  – число рядів у секції;

$R$  і  $NTU_2$  – відношення водяних еквівалентів теплоносіїв і число одиниць переносу теплоти в секції [1, 2],  $R = \frac{W_{н}}{W_{Г}}$ ,  $NTU_2 = \frac{K \cdot F}{W_{н}}$ ,

$a = n \cdot (1 - \exp(-NTU_2/n))$ ;

$K$  і  $F$  – коефіцієнт та площа теплопередачі в поточній секції.

б) для наступних рядів поточної секції ( $i = 2 \mathbf{K} n$ ):

$$\overline{t_{Г\bar{x}ik}} = \frac{t_{Г1k} - t_{Г\bar{x}ik}}{t_{Г1} - t_{н1}} = e^{-R \cdot a \cdot \bar{x}} \cdot \omega_{ik}, \quad i = 2 \mathbf{K} n, \quad (4)$$

$$\omega_{ik} = \left(1 - \frac{a}{n}\right) \cdot \omega_{(i-1),k} + R \cdot a \cdot \left(\frac{a}{n}\right) \int_0^{\bar{x}} \omega_{(i-1),2} d\bar{x}, \quad i = 2 \mathbf{K} n, \quad (5)$$

$$\omega_{1k} = e^{R \cdot a \cdot \bar{x}} \cdot \overline{t_{Г\bar{x}1k}}.$$

Слід зазначити, що в схемі з протитоком локальні температури середовища, що гріє, можуть бути визначені тільки після розрахунку

параметрів  $\bar{\Phi}_k$  шляхом розв'язку системи рівнянь (1)–(5) методом послідовних наближень при значенні відносної координати  $\bar{x} = 1$ :

$$\bar{\Phi}_k^q = \bar{\Phi}_{(k-1)}^{(q-1)} + \frac{1}{n} \cdot \sum_{i=1}^n t_{\Gamma(\bar{x}=1)ik}^{(q-1)}, \quad \bar{\Phi}_0^0 = \bar{\Phi}_0^q = 0, \quad (6)$$

де  $q$  – індекс поточного наближення.

Температура середовища, що гріє, перед поточною секцією:

$$t_{\Gamma 1k} = t_{\Gamma 1ik} = t_{\Gamma 1} - \bar{\Phi}_{(k-1)} \cdot (t_{\Gamma 1} - t_{\Gamma 1}), \quad (7)$$

Підсумкова ефективність теплообмінника і температури середовища на виході:

$$P = \frac{\bar{\Phi}_z}{R}; \quad t_{\Gamma 2} = t_{\Gamma 1} + P \cdot (t_{\Gamma 1} - t_{\Gamma 1}); \quad t_{\Gamma 2} = t_{\Gamma 1} - P \cdot R \cdot (t_{\Gamma 1} - t_{\Gamma 1}). \quad (8)$$

Як видно з аналізу вищенаведеного алгоритму, розв'язок системи рівнянь складний, супроводжується ітераціями, рекурентним обчисленням інтегральних перетворень. Для багатоходових секцій розв'язок ще більш ускладнюється.

Узагальнена схема багатоходового багатосекційного теплообмінника зі змішаною схемою плинку теплоносіїв і протитоковим включенням ходів представлена на рис. 2.

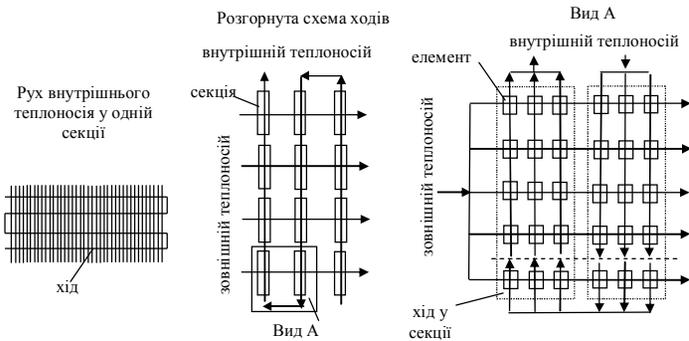


Рисунок 2 – Узагальнена схема руху теплоносіїв у системі теплообмінників

Для спрощення пошуку розв'язку авторами була запропонована методика дискретного розрахунку, де елементами, з яких скомпонований теплообмінник (див. рис. 2, вид А), є найпростішими схемами однократного перехресного плинку з повним перемішуванням обох теплоносіїв по ходу. Слід

зазначити, що більшість традиційних підходів до дискретного розрахунку теплообмінників передбачає розбивку поверхні на велике число кінцевих різностей, де, як правило, не враховуються особливості руху середовищ в елементах. У запропонованій авторами методиці ефективність кожного перехресноплинного елемента з рис. 2 та температури теплоносіїв на виході з елементів будуть визначатися в такий спосіб:

$$P_e = \frac{1}{\frac{1}{1 - e^{-NTU_{2e}}} + \frac{R_e}{1 - e^{-R_e \cdot NTU_{2e}}} - \frac{1}{NTU_{2e}}}; \quad (9)$$

$$t_{\Gamma 2e} = t_{\Gamma 1e} + P_e (t_{\Gamma 1e} - t_{\Gamma 1e}) \text{ і } t_{\Gamma 2e} = t_{\Gamma 1e} - P_e R_e (t_{\Gamma 1e} - t_{\Gamma 1e}), \quad (10)$$

де "e" – індекс, який позначає, що параметри визначаються в елементі; "1" – вхід середовища, "2" – вихід середовища.

При складанні алгоритму розв'язку системи рівнянь (9)–(10), що записуються для кожного елемента з рис. 2, враховується схема взаємного з'єднання елементів і перемішування теплоносія між ходами та на виході з апарата. По ходу руху теплоносіїв обчислюються втрати тисків від тертя та місцеві втрати. Ефективність кожного елемента визначається з урахуванням розходження властивостей теплоносіїв і матеріалів стінок труб, параметрів, що характеризують теплопередачу. Тому що схема руху протитокова, то для уточнення ефективності елементів використовується інтервально-ітераційний метод. У підсумку ефективність всього апарата визначається таким чином:

$$P = \frac{1}{R} \frac{t_{\Gamma 1} - t_{\Gamma 2}}{t_{\Gamma 1} - t_{\Gamma 1}}, \quad (11)$$

де  $t_{\Gamma 2} = \frac{1}{n} \sum_{i=1}^{n_r} t_{\Gamma 2si}$ ;  $n$  – число рядів труб у секції;  $t_{\Gamma 2si}$  – температури

середовища, що гріє, на виході з елементів останнього ходу останньої секції.

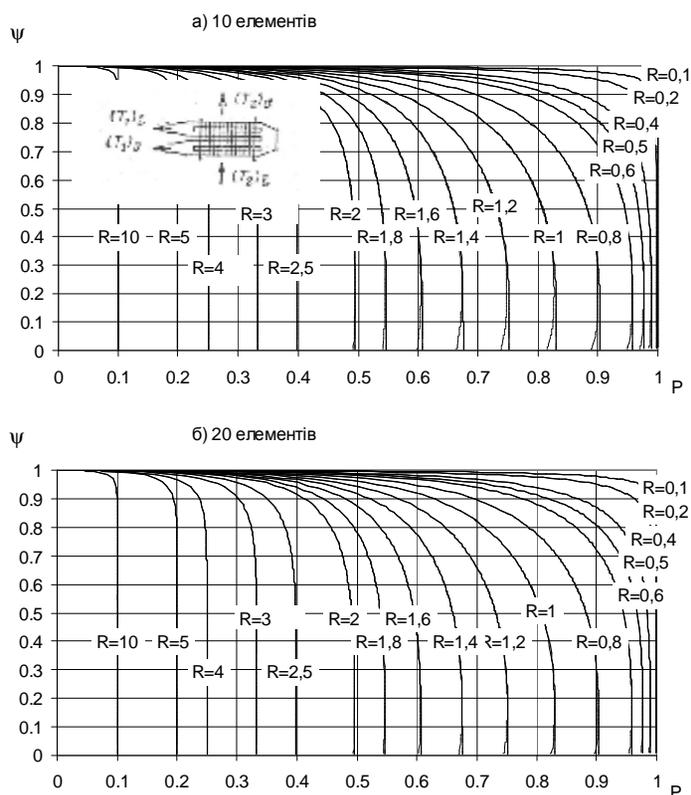
Відомий поправочний коефіцієнт до середньологарифмічного температурного напору при протитоку визначається методом послідовних наближень з урахуванням залежності:

$$\psi = \frac{\Delta t_{сеп}}{\Delta t_n} = \frac{1}{NTU_2 \cdot \delta}; \quad \delta = \frac{R-1}{\ln \frac{1-P}{1-PR}} (R \neq 1); \quad \delta = \frac{1-P}{P} (R=1). \quad (12)$$

Авторами були зіставлені результати розрахунку по розробленим аналітичному алгоритму до рис. 1 і дискретному алгоритму для відповідної

одноходової багатосекційної схеми з рис. 2 при однакових коефіцієнтах теплопередачі та властивостях теплоносіїв у секціях.

Результати розрахунку коефіцієнта  $\psi$  для одноходової схеми з двома дворядними секціями приведено на рис. 3 (безперервна крива – аналітичний алгоритм, переривиста крива – дискретний алгоритм).



1 – теплоносій, що гріє; 2 – теплоносій, що нагрівається;  $i$  – вхід;  $o$  – вихід

Рисунок 3 – Порівняння результатів розрахунку

У результаті проведеного аналізу відзначено, що при числі дискретних елементів у ряді більше  $40/z$  (для кількості ходів 4 і менше) і 10 при більшому числі ходів, розходження в результатах практично відсутні. При використанні традиційного підходу кінцевих різниць для досягнення такої ж точності необхідно розбивати поверхню у  $10\div 100$  разів більше. Однак, як правило,

кількість елементів у ряді в апараті визначається кількістю ребер, або ступінню турбулентного перемішування.

Методика дискретного розрахунку була апробована для апаратів повітряного охолодження, калориферів, повітроохолоджувачів компресорів, теплообмінників систем тепlopостачання [3-6] та ін.

Результати розрахунків параметрів  $P$  і  $\psi$  за розробленими методиками для деяких окремих випадків були зіставлені з номограмами з [2] і відзначено їхній збіг, що говорить про вірогідність розроблених математичних моделей, методик та алгоритмів. На відміну від формул і номограм з [2], створена методика дозволяє розраховувати характеристики апаратів з довільною кількістю рядів, секцій і ходів, у тому числі – і з дискретним розподілом параметрів теплопередачі.

**Висновки.** За допомогою розроблених методів та засобів можна одержати розподіл локальних температурних характеристик та ефективність складних багатходових та багатосекційних теплообмінників зі змішаною схемою плинину. Створена математична модель дозволяє: визначити теплову продуктивність теплообмінників з довільним числом рядів, ходів і секцій з урахуванням особливості компоновання, проводити системний аналіз апаратів, їх ефективності на різних режимах роботи, у тому числі з урахуванням умов експлуатації. Результати моделювання для деяких окремих випадків були зіставлені з відомими залежностями і номограмами і відзначено, що вони практично співпадають. Тому розроблені узагальнені математичні моделі, методики і комплекси алгоритмів можна застосовувати для будь-яких аналогічних багатходових багатосекційних теплообмінників зі змішаною схемою плинину теплоносіїв.

**Список літератури:** 1. Кейс В.М., Лондон В.М. Компактные теплообменники: Пер. с англ. В.Я. Сидорова / Под ред. Ю.В. Петровского.– М.: Энергия, 1967. – 223 с. 2. Справочник по теплообменникам: Пер. с англ.: В 2 т. / Под ред. Б.С. Петухова, В.К. Шикова. – М.: Энергоатомиздат, 1987.– Т.1. – 560 с. 3. Ганжа А.Н., Марченко Н.А. Анализ эффективности аппаратов воздушного охлаждения // Двигатели внутреннего сгорания. – 2005. – №1. – С. 81–85. 4. Ганжа А.Н., Марченко Н.А. Определение ресурса воздухоохлаждателей компрессорных установок с использованием имитационного моделирования // Двигатели внутреннего сгорания. – Харьков: НТУ "ХПИ", 2009. – № 2. – С. 12–16. 5. Братута Э.Г., Ганжа А.Н., Марченко Н.А. Анализ эффективности калориферного подогреву воздуха у системах опалення, вентиляції та кондиціонування // 36. наук. праць 5-ї Міжнародної науково-практичної конференції "Проблеми економії енергії".– Львів, НУ "Львівська політехніка", 23-24 жовтня 2008. – С.274–278. 6. Ганжа А.Н., Братута Э.Г., Марченко Н.А. Уточненная методика определения тепловой производительности системы отопления с учетом неравномерности распределения параметров // Интегрированные технологии та энергосбережения. – 2009. – № 2. – С. 66–70.

Поступила в редакцию 21.12.09

**С. В. КОВАЛЕНКО**, ст. преподаватель НТУ «ХПИ»  
**С. Н. КОВАЛЕНКО**, канд. техн. наук, ХНТУСХ им. П.Василенко,  
**Н. В. БИРЮКОВА**, магистрант НТУ «ХПИ»

### АНАЛИЗ МЕТОДОВ ПОИСКА СХОДНЫХ ИЗОБРАЖЕНИЙ СРЕДИ БОЛЬШИХ КОЛЛЕКЦИЙ ГРАФИЧЕСКИХ ФАЙЛОВ

У роботі розглядаються способи рішення задачі пошуку схожих зображень у колекціях зображень. Технологія пошуку реалізується на базі зіставлення набору візуальних примітивів і визначенням кількісної оцінки близькості зображень за значеннями примітивів.

В работе рассматриваются способы решения задачи поиска похожих изображений в коллекциях изображений. Технология поиска реализуется на базе сопоставления набора визуальных примитивов и определением количественной оценки близости изображений по значениям примитивов.

The paper discusses various questions connected with finding similar images in the collections of images. Search technology is implemented on the basis of comparison set of visual primitives and the definition of quantitative estimates of how close the values of image primitives.

**Введение.** В наше время наблюдается постоянный рост объема обрабатываемой информации. Количество изображений самого разного характера постоянно растет. Интернет и цифровые библиотеки дают доступ к огромному количеству информации.

Методы распознавания образов в настоящее время из-за отсутствия эффективных универсальных алгоритмов применяются в узких предметных областях. Современная универсальная технология доступа к коллекциям изображений связана с сопоставлением изображению набора визуальных примитивов (характеристик цвета, формы, текстуры) и определением количественной оценки близости изображений по значениям примитивов.

Визуальные примитивы [1] – это характеристики изображения, которые автоматически вычисляются по визуальным данным, позволяют эффективно индексировать их и обрабатывать запросы с использованием визуальных свойств изображения. Поиск по образцу, сгенерированный из визуальных примитивов, невелик по размеру в сравнении с самим изображением и удобен для организации поиска. Запросным механизмом является поиск по образцу. В этом случае система отыскивает изображения, визуально похожие на предоставленный образец. Поиск на таком уровне абстракции не предполагает идентификацию объектов. Тем не менее, метод поиска по образцу на основании визуальных примитивов представляется на сегодняшний день достаточно эффективным и универсальным средством доступа к коллекциям оцифрованных изображений.

**Общая постановка задачи.** Для организации электронных библиотек, связанных с визуальными данными, требуется использовать методы создания поисковых образов, отражающих визуальное содержание изображений.

Задача состоит в необходимости отыскать в большой коллекции графических файлов изображения, не являющиеся идентичными, однако обладающие некоторой степенью схожести. В зависимости от уровня этой степени найденные изображения могут походить друг на друга больше или меньше. Похожие изображения группируются в кластеры для дальнейшей работы с ними [3]. Подобный идейный подход реализован в экспериментальном сервисе Google Similar Images, который позволяет находить в интернете по запросу не текстовую информацию, а и графическую.

**Методы решения поставленной задачи.** Важность для человека цветового восприятия изображения объясняет значение методов поиска визуальной информации на основании схожести с образцом по цветовым характеристикам. Метод цветových гистограмм – наиболее популярный из методов, использующих цветовые характеристики для индексирования изображений. Идея метода цветových гистограмм для индексирования и сравнения изображений сводится к следующей последовательности действий. Все множество цветов разбивается на набор непересекающихся, полностью покрывающих его подмножеств  $V_i$ ,  $0 \leq i < N$ . Для изображения формируется гистограмма, отражающая долю каждого подмножества цветов в общей цветовой гамме изображения. Для сравнения гистограмм вводится понятие расстояния между ними. Известны различные способы построения и сравнения цветových гистограмм, отличающиеся между собой изначальной цветовой схемой, размерностью гистограммы и определением расстояния между гистограммами.

В данной статье рассмотрены методы, использующие разные способы квантования множества цветов и вычисление расстояния между гистограммами.

**Разбиение цветов, представленных в модели RGB по яркости.** В базовой палитре  $V_i$  определяется как множество цветов

$$C : C \in V_i \Leftrightarrow \frac{i}{N \cdot I_{\max}} \leq I(C) < \frac{i+1}{N \cdot I_{\max}}, \quad (1)$$

где  $I(C)$  – интенсивность цвета  $C$ , нормализованная таким образом, что  $0 \leq I(C) < I_{\max}$ .

Интенсивность вычисляется по формуле:

$$I(C) = 0.3R(C) + 0.59G(C) + 0.11B(C), \quad (2)$$

где R, G и B – красная, зеленая и синяя компоненты цвета  $C$ ,  $I_{\max} = 256$ ,  $0 \leq I(C) < 256$ . Значение  $N$  является количеством каналов, на которые разбивается цветовая составляющая.

Для сравнения гистограмм вводится понятие расстояния между ними – сумма модулей разности соответствующих элементов гистограмм. Некоторое усовершенствование метода достигается при вычислении расстояния на основании поэлементного сравнения гистограмм с учетом соседних элементов. При этом для каждого элемента гистограммы первого изображения вычисляется не одна, а три разности:

$$\begin{aligned} R_1[i] &= |H_1[i] - H_2[i-1]|, \\ R_2[i] &= |H_1[i] - H_2[i]|, \\ R_3[i] &= |H_1[i] - H_2[i+1]|. \end{aligned} \quad (3)$$

Для  $i=0$  и  $i=N$  вместо невычислимых разностей подставляются заведомо большие значения.

Этот метод построения гистограмм наиболее эффективен для черно-белых полутоновых изображений. Для цветных RGB-изображений лучшие результаты дает другой способ – разбиение RGB-цветов по прямоугольным параллелепипедам.

**Разбиение по параллелепипедам.** Цветовое RGB-пространство рассматривается как трехмерный куб, каждая ось которого соответствует одному из трех основных цветов. При таком рассмотрении любой цвет RGB-изображения может быть представлен точкой куба. Для построения цветовой гистограммы каждая сторона делится, например, на  $n$  ( $n=4$ ) равных интервалов, соответственно RGB-куб делится в данном случае на  $N$  ( $N=64$ ) прямоугольных параллелепипедов, при этом  $V_i$  – множество цветов, все компоненты которых попадают в определенные интервалы. Гистограмма изображения отражает распределение точек RGB-пространства, соответствующих цветам пикселей изображения, по параллелепипедам.

В качестве расстояния между гистограммами используется покомпонентная сумма модулей разности между ними. Несмотря на предельную простоту подхода, он показывает довольно стабильные результаты.

**Техника квадродеревьев.** Более точное сравнение изображений достигается с помощью техники квадродеревьев, когда методы вычисления и сравнения цветных гистограмм применяются не ко всему изображению, а к его четверти (одной шестнадцатой и т. д.). Целесообразность ее применения определяется значением для пользователя расположения на картинке-образце определенных цветных областей.

**Пространственное сегментирование.** Сегментирование изображения может осуществляться автоматически, когда выделяются области с некими общими свойствами – одинаковыми или сильно схожими значениями того или иного примитива. Полученные в результате области характеризуются расположением на изображении и размерами.

Определение границ объектов изображения выполняется с помощью фильтрации изображения: цветное изображение переводится в черно-белое полутоновое и сглаживается, осуществляется пространственное дифференцирование – вычисляется градиент функции интенсивности в каждой точке изображения и, наконец, подавляются значения меньше установленного порога. За основу взят метод Собеля, использующий для вычисления градиента первого порядка функции интенсивности – специальные ядра, известные как «операторы Собеля». Ядра применяются к каждому пикселу изображения: он помещается в центр ядра, и значения интенсивности в соседних точках умножаются на соответствующие коэффициенты ядра (см. рисунок), после чего полученные значения суммируются. X-оператор Собеля, примененный к матрице исходного изображения, дает величину горизонтальной составляющей градиента интенсивности в центральной точке этой матрицы, а Y-оператор Собеля дает величину вертикальной составляющей градиента.

В результате получается массив чисел, характеризующих изменения яркости в различных точках изображения. На следующем шаге выполняется операция сравнения с пороговым значением и определяется положение элементов изображения с наиболее сильными перепадами яркости.

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

а) X-оператор Собеля

б) Y-оператор Собеля

Операторы Собеля

В качестве основного порога берется средняя для изображения величина градиента. Для достаточно большого изображения с малым числом точек, обладающих сильным перепадом яркости, данной пороговой величины недостаточно. Это связано с тем, что для изображений подобного вида оказывается весьма сильным влияние шума. Для устранения этой проблемы для каждой точки изображения рассчитывается величина  $S_{local}$ , равная средней величине градиента в области  $3 \times 3$  вокруг анализируемой точки. Пороговое условие в данном случае выглядит следующим образом:

$$(G_{i,j} \geq S_{mid}) \text{ AND } (S_{local} \geq S_{mid}) \quad (4)$$

где  $S_{mid}$  – средняя для изображения величина градиента.

В результате обработки получается бинарная матрица, где единицам соответствуют точки со значительным перепадом яркости, нулям – все остальные.

В дальнейшем при сегментации изображения в бинарной матрице единицами представлены точки, принадлежащие искусственно утолщенным на предыдущем этапе границам объектов. Для выделения границы одного объекта в матрице по определенному алгоритму отыскивается элемент, равный единице, не отнесенный ранее ни к какому другому объекту. На следующем шаге считается, что все соседние элементы, равные единице, также принадлежат этому объекту. Для выделения точек внешнего контура используется обход полученного объекта по внешней его стороне, начиная с нижней левой точки объекта и заканчивая ею же. В результате получаем массив точек, образующий замкнутый контур объекта. Контур [2] – граница объекта, которая представляет собой замкнутую последовательность точек  $(x_s, y_s)$ , где  $1 \leq s \leq N$ . Из него равномерно выбирается 128 точек  $(x_s, y_s)$ , где  $1 \leq s \leq 128$ , которые используются для вычисления предназначенных для индексирования характеристик формы.

**Кластерный анализ.** После применения описанных выше методов осуществляется кластеризация схожих изображений. Задача кластерного анализа заключается в том, чтобы на основании данных о схожести изображений разбить их множество  $G$  на  $m$  кластеров (подмножеств)  $Q_1, Q_2, \dots, Q_m$  так, чтобы каждый объект  $G_j$  принадлежал одному и только одному подмножеству разбиения и чтобы объекты, принадлежащие одному и тому же кластеру, были сходными, в то время как объекты, принадлежащие разным кластерам, были разнородными.

**Заключение.** Таким образом, применяя различные методы сравнения изображений, можно найти похожие изображения в коллекции графических файлов. Группируя найденные изображения в кластеры, становится возможной работа с группами похожих изображений внутри большой группы файлов. Внутри определенного кластера, размеры которого во много раз меньше общей совокупности графических файлов коллекции, возможна дальнейшая обработка.

**Список литературы:** 1. Байгарова Н.С., Бухштаб Ю.А., Евтеева Н.Н., Корягин Д.А. Некоторые подходы к организации содержательного поиска изображений и видеoinформации.– Москва: Институт прикладной математики им. Келдыша РАН, 2002.– 24 с. 2. Гончаров А., Мельниченко А. Pseudometric Approach to Content Based Image Retrieval and Near Duplicates Detection.– РОМИП, 2008.– 34 с. 3. Пономаренко Н.Н., Лукин В.В., Абрамов С.К. Устойчивый поиск изображений по полному и тематическому подобию с использованием многопараметровой классификации.– Харьков: «ХАИ», 2003.– 13с.

*Поступила в редколлегию 27.11.09*

УДК 621.3

**Н. И. БЕЗМЕНОВ**, канд. техн. наук, профессор НТУ «ХПИ»,  
**С. В. КОВАЛЕНКО**, старший преподаватель НТУ «ХПИ»,  
**М. А. ТКАЧЕВ**, магистрант НТУ «ХПИ»

## ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РАСПОЗНАВАНИЯ ГРАФИЧЕСКИХ ОБРАЗОВ

Пропонується використовувати нейронні мережі на основі базисно радіальних функцій як метод для розпізнавання графічних образів. Продемонстрована працездатність мережі на прикладі розпізнавання цифр.

Предлагается использовать нейронные сети на основе базисно радиальных функций в качестве метода для распознавания графических образов. Продемонстрирована работоспособность сети на примере распознавания цифр.

It is proposed to use neural network based on radial basic functions as a method for recognition of graphic images. The capacity of network is retined on the example of recognition of numbers.

**Введение.** Одной из актуальных проблем современных информационных технологий в настоящее время остается распознавание образов, в частности, графически представленных символов. Для решения данной задачи используются ряд алгоритмов, которые можно условно разделить на три категории:

- алгоритмы, основанные на сравнении шаблонов;
- алгоритмы, основанные на методах теории решений;
- алгоритмы, использующие нейронные сети.

Все вышеперечисленные группы алгоритмов имеют, в целом, одинаковую структуру и содержат такие блоки:

- блок обучения системы на наборе образов, подобным тем, которые будут распознаваться системой в будущем. По завершению обучения, его результаты сохраняются;
- блок распознавания, который, используя информацию, полученную на этапе обучения, относит незнакомый образ к одному из классов с помощью определенного математического аппарата (классификатора).

Алгоритмы, основанные на сравнении шаблонов, из-за своей негибкости (привязанности к шрифту, масштабу, наклону символов) не подходят для общего решения данной задачи. Однако в силу своей простоты и малым вычислительным затратам, они (с определенными модификациями) достаточно часто используются при решении отдельных задач распознавания [1].

Метод нейронных сетей, использующий некоторые принципы искусственного интеллекта, широко применяется в большинстве

современных систем распознавания символической информации. Среди его преимуществ стоит упомянуть универсальность и незначительные требования к проектированию алгоритмов. Недостатками данного метода являются значительный объем вычислительных ресурсов, необходимых для организации процесса обучения, а также плохая предсказуемость результатов распознавания. Наибольшим преимуществом таких методов является высокая точность распознавания.

Алгоритмы, основанные на методах теории решений, во много раз упрощают процесс обучения, но их точность значительно уступает точности нейросетевых алгоритмов. Важным преимуществом данных алгоритмов является представление символов в векторной форме (в виде набора характеристик), в то время как большинство нейросетевых алгоритмов оперируют с растровым изображением. При использовании некоторых классификаторов теории решений, имеется возможность добиться высокой степени предсказуемости результатов распознавания.

Далее рассматривается метод распознавания графических образов на основе нейронных сетей. В качестве архитектуры нейросети была выбрана сеть, основанная на радиально базисных функциях.

**Введение в RBF-сети.** Базовая архитектура сетей на основе радиальных базисных функций (radial basis function network - RBF), или RBF-сетей, предполагает наличие двух слоев, выполняющих совершенно различные функции. Входной слой состоит из сенсорных элементов, которые связывают сеть с внешней средой. Второй слой является единственным скрытым слоем сети и выполняет нелинейное преобразование входного пространства в скрытое.

Математическую основу функционирования радиальных сетей составляет теорема Т. Ковера о распознаваемости образов [3], в соответствии с которой нелинейные проекции образов в некоторое многомерное пространство могут быть линейно разделены с большей вероятностью, чем при их проецировании в пространство с меньшей размерностью.

Простейшая нейронная сеть радиального типа функционирует по принципу многомерной интерполяции, состоящей в отображении  $p$  различных входных векторов  $x_i, i = 1, 2, \dots, p$  из входного  $N$ -мерного пространства во множество из  $p$  чисел  $d_i, i = 1, 2, \dots, p$ . Для реализации этого процесса необходимо использовать  $p$  скрытых нейронов радиального типа и задать такую функцию отображения  $F(x)$ , для которой выполняется условие интерполяции  $F(x_i) = d_i$ .

Использование  $p$  скрытых нейронов, соединяемых связями с весами с выходными линейными нейронами, означает формирование выходных сигналов сети путем суммирования взвешенных значений соответствующих базисных функций. Рассмотрим радиальную сеть с одним выходом и  $p$

обучающими парами  $(x_i, d_i)$  и будем считать, что координаты каждого из  $p$  центров узлов сети определяются одним из векторов  $x_i$ , т.е.  $c_i = x_i$ . В этом случае взаимосвязь между входными и выходными сигналами сети может быть определена системой уравнений, линейных относительно весов. Эта система в матричной форме имеет вид:

$$j \cdot w = d, \quad (1)$$

где  $j_{ji}$  – определяет радиальную функцию с центром в точке  $x_i$  с вынужденным вектором  $x_j$ ,  $w = [w_1, w_2, \dots, w_p]^T$  и  $d = [d_1, d_2, \dots, d_p]^T$ .

Доказано [2], что для ряда радиальных функций в случае  $x_i \neq x_j \neq \mathbf{K} x_p$  квадратная интерполяционная матрица  $j$  является невырожденной и при этом неотрицательно определенной. Поэтому существует решение уравнения (1) в виде:

$$w = (j^T j)^{-1} j^T d, \quad (2)$$

что позволяет получить вектор весов выходного нейрона сети.

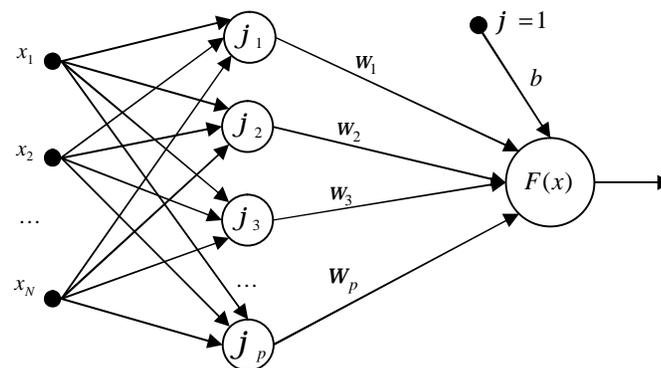


Рис. 1. Обобщенная структура радиальной сети

Чаще всего в качестве радиальной функции применяется функция Гаусса. При размещении ее центра в точке  $c_i$  она может быть определена в сокращенной форме:

$$j(x) = j \left( \|x - c_i\| \right) = \exp\left(\frac{-\|x - c_i\|^2}{2s_i^2}\right). \quad (3)$$

В этом выражении  $S_i$  - параметр, от значения которого зависит ширина функции.

На рис. 1 в общем виде изображена структура RBF-сети. Это сеть с двухслойной структурой, в которой только скрытый слой выполняет нелинейное отображение, реализуемое нейронами с базисными радиальными функциями. Выходной нейрон, как правило, линеен, а его роль сводится к взвешенному суммированию сигналов, поступающих от нейронов скрытого слоя.

**Пример работы сети.** В качестве примера работы сети попытаемся различить символы «5» и «4», изображения которых представлены на рис. 2.

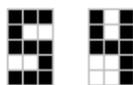


Рис. 2. Тестовые изображения

В этом случае сеть принимает вид, представленный на рис. 1. Имеется всего два образа. Количество нейронов скрытого слоя будет равняться двум. Количество сенсорных элементов, составляющих входящего вектора образа, равняется количеству элементов в изображении – 15 в данной задаче.

Запишем пару векторов обучающей выборки, воспользовавшись следующей методикой: переводя изображения в вектора (построчно перемещаясь по квадратам), будем ставить в соответствие заполненной области единицу, а пустой – 0. В результате получаем такие векторы:

$$x_1 = \{1,1,1,1,0,0,1,1,1,0,0,1,1,1,1\}, \quad (4)$$

$$x_2 = \{1,0,1,1,0,1,1,1,1,0,0,1,0,0,1\}. \quad (5)$$

В качестве центров примем соответственно значения  $c_1 = x_1$  и  $c_2 = x_2$ , в результате чего получим следующий вектор  $d$ :

$$d = [0,1]^T. \quad (6)$$

Каждый его элемент является желаемым откликом сети для соответствующего вектора образа.

Искомый вектор  $w$  имеет вид  $w = [w_1, w_2, b]^T$ , где  $w_1$  и  $w_2$  – весовые коэффициенты соответствующих образов,  $b$  – порог выходного нейрона.

Пусть

$$j_{ji} = \langle \|x_j - c_i\| \rangle, j = 1,2; i = 1,2. \quad (7)$$

Тогда, подставляя векторы обучающих выборок (4) и (5) в (7), получаем матрицу  $j$ :

$$j = \begin{bmatrix} 1 & 0.018 & 1 \\ 0.018 & 1 & 1 \end{bmatrix}. \quad (8)$$

Теперь, подставляя матрицу  $j$  вида (8) и вектор желаемых откликов  $d$  вида (6) в (2), получим веса  $w$ :

$$w = \begin{bmatrix} -0.416 \\ 0.811 \\ 0.27 \end{bmatrix}, \quad (9)$$

что и завершает построение архитектуры сети.

Функция активации для данной задачи имеет следующий вид:

$$F(x) = \sum_{i=1}^2 w_i j_i (\|x - c_i\|) + b. \quad (10)$$

В качестве проверки работоспособности сети, попытаемся получить желаемые отклики на соответствующие образы. Подставляя векторы образов  $x_1$  и  $x_2$  в функцию активации (10), получим соответственно значения  $[-0.131, 1.073]$ , которые с некоторой погрешностью говорят о способности сети к классификации образов.

**Выводы.** Таким образом, в результате проведенных исследований, можно утверждать о способности сети к классификации векторов образов на основе базисно радиальных функций. Тестовый пример показывает положительные результаты распознавания символов печатного текста.

**Список литературы:** 1. Ватолин Д.С. Алгоритмы сжатия изображений. – МГУ, 2002. – С. 30- 32. 2. Micchelli C.A. "Interpolation of scattered data: Distance matrices and conditionally positive definite functions" // Constructive Approximation, 1986, vol. 2, p. 11-22. 3. Powell M.J.D. "Radial basis functions for multivariable interpolation: A review" // IMA Conference on Algorithms for the Approximation of Functions and Data, 1985, p. 143-167, RMCS, Shrivvenham, England.

Поступила в редколлегию 11.12.09

**Т. В. ГОРБАЧ**, аспирант ХНУРЭ,  
**Я. В. СВЯТКИН**, аспирант ХНУРЭ,  
**И. Ю. ШУБИН**, канд. техн. наук, доц., проф. кафедры ПОЭВМ ХНУРЭ

### МЕТОДЫ ИНТЕГРАЦИИ УЧЕБНЫХ МАТЕРИАЛОВ ДЛЯ ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Запропоновано модель синтаксичного аналізу у задачах анотування та реферування повнотекстової інформації, з використанням алгебри скінчених предикатів та предикатних операцій. Проведено аналіз існуючих методів анотування та реферування повнотекстової інформації. Запропоновано використовувати при анотуванні та реферуванні статистичні методи та підхід, що розроблений, разом.

Предложена модель синтаксического анализа в задачах аннотирования и реферирования полнотекстовой информации, основанная на использовании конечных предикатов и предикатных операций. Проведено анализ существующих методов аннотирования и реферирования полнотекстовой информации. Предложено использовать при аннотировании и реферировании статистические методы и предложенную модель вместе.

The model of the syntactic analysis in problems of annotation and abstracting is offered to the text-through information, based on use of final predicates and predicate operations. It is lead the analysis of existing methods of annotation and abstracting the text-through information. It is offered to use at annotation and реферировании statistical methods and the offered model together.

**Введение.** Наиболее часто встречающейся проблемой, которая стоит перед авторами учебных курсов при компьютерном (дистанционном) обучении, является интеграция разнородных учебно-методических материалов в единый курс обучения. Актуальной является задача создания инструментальных систем поддержки интеграции учебно-методических материалов, к которым предъявляются жесткие требования, так как эффективность обучения существенно зависит от формы и качества предоставления учебных материалов. В основу подобных систем закладывается распределенная модель хранения информации.

Основу содержания современных учебных курсов должны составлять осваиваемые способы деятельности, а не конкретное предметное содержание [1]. Данное обстоятельство обуславливает необходимость многопоточной серверной реализации тех функций системы, которые связаны с проверкой ответов обучаемых и предоставлением доступа к системе и к базам данных учебных курсов.

**Анализ последних достижений.** Автоматизированные обучающие системы (АОС) [2] представляют собой программно-технические комплексы, включающие в себя методическую, учебную и организационную поддержку процесса обучения, проводимого на базе информационных технологий.

Существующие АОС разбивают на четыре функциональных класса [3]:

- информационные системы, содержащие большой объем методического материала и слабую контролирующую часть;
- контролирующие системы, содержащие в качестве обучающего материала только справочный материал и имеющие мощный контролирующий блок;
- моделирующие системы, моделирующие поведение интересующего нас объекта и предназначенные для выполнения лабораторных работ, когда необходимо исследовать поведение его в зависимости от внешних воздействий при заданных параметрах;
- тренажеры, необходимые для обучения каким либо действиям, например, управлению автомобилем или тренировки персонала в условиях чрезвычайных ситуаций.

Информационные системы в зависимости от характера учебных занятий можно разбить на следующие типы [4]:

- лекторские;
- ассистентские;
- репетиторские;
- справочно-консультирующие системы.

Каждый тип систем отличается способом представления предмета обучения и формой обучения, которые имитируют подобную деятельность преподавателя.

Весьма распространенным типом АОС среди замкнутых систем являются имитационные автоматизированные обучающие системы. Здесь функции ведущего "элемента" выполняет фактор моделирования реальной ситуации в той или иной сфере предметной области. Элемент с обратной связью в виде реакции ученика на предъявляемый АОС учебный материал является основой непрерывного взаимодействия системы "АОС – Обучаемый", т. к. то или иное воздействие на систему со стороны пользователя ведет сразу к ответной реакции со стороны обучающей системы.

**Постановка задачи и цель работы.** Разработка математического обеспечения для создания универсальных компьютерных программ учебного назначения (КПУН), входящих в состав инструментальной системы поддержки интеграции учебно-методических материалов для построения интеллектуальной компьютерной обучающей системы (ИКОС) с накоплением информации, управляемой правилами. Обновление базы знаний КПУН производится с помощью инструментальной системы по мере надобности, в том числе и по результатам апостериорной статистической обработки матрицы результатов измерения знаний достаточно большого количества обучаемых. Использование такого подхода к проектированию инструментальных систем интеграции различных материалов в учебный курс позволяет достичь следующих целей:

1. Реализовать индивидуальный подход к объекту обучения. ИКОС является особым классом обучающих систем, в структуру которых закладываются механизмы адаптации к конкретному объекту обучения. В состав ИКОС включаются вычислительные, классифицирующие, контрольно-диагностирующие средства, позволяющие оценивать параметры объекта обучения и вырабатывать оптимальную стратегию и тактику обучения на каждом этапе.
2. Проводить обучение в режиме реального времени. Процедуры адаптации должны выполняться с высокой скоростью, поскольку любые технологические задержки в работе системы можно рассматривать как возмущающие воздействия, влекущие за собой отклонения процесса обучения от оптимального.

**Решение задачи.** Учебный курс представляется в виде направленного графа, в котором узлами служат документы специального вида – слайды, а ребрами – переходы между ними. Из каждого узла обучаемый может попасть в один из нескольких других узлов, непосредственно связанных с ним ребрами. Какой переход будет выбран, решает система на основе данных об обучаемом и текущего состояния учебного процесса. В первую очередь при решении используются протокол работы и модель обучаемого. Протокол представляет собой файл, в который автоматически заносится информация о всех действиях, которые обучаемый выполнял за терминалом компьютера. На основе анализа протокола, при необходимости, можно скорректировать модель обучаемого. Модель обучаемого должна содержать ответы на вопросы не только о том, что знает и умеет делать конкретный обучаемый, но и о том, к какому психологическому типу он относится и чего достиг за время обучения.

Некоторые узлы курса могут быть помечены как обязательные для посещения. Стартовый узел выбирается на основе уровня подготовки обучаемого, выявленного с помощью блока вступительного тестирования. Выбор конкретного перехода (направления дальнейшего развития учебного процесса) осуществляется на основе правил, связанных с каждым из ребер перехода.

При этом инструментальная система позволяет разработчикам вносить коррективы в эти правила с помощью средств визуального редактирования без необходимости ввода текста правил вручную.

Информация в узлах курса обучения должна быть разбита на небольшие смысловые части – слайды. Слайды могут быть двух типов: интерактивные и информационные. Информационные слайды состоят только из объектов отображения текстовой и графической информации, воспроизведения аудио- и видеофрагментов и кнопок управления ходом учебного процесса. Интерактивные слайды, кроме перечисленных объектов, могут также содержать объекты, реализующие различные виды тестовых заданий для организации обратной связи с обучаемым и оперативного контроля усвоения

знаний, а также для приобретения обучаемым необходимых навыков. В текущей версии данной инструментальной системы применительно к интерактивным слайдам используется биномиальная модель педагогического теста. В рамках этой модели результат выполнения каждого задания оценивается по бинарной шкале: 1, если испытуемый успешно выполнил задание, содержащееся в данном интерактивном слайде; 0 – в противном случае. Таким образом, в учебном курсе каждый интерактивный слайд имеет как минимум два исходящих ребра: одно соответствует успешному выполнению тестового задания, второе – неправильному решению задания. При таком подходе, количество интерактивных слайдов в блоке интерактивного обучения, а также количество вопросов в блоке заключительного тестирования определяется следующими параметрами:  $N$  – искомое количество интерактивных слайдов;  $p$  – вероятность успешного выполнения тестового задания (истинный балл обучаемого, характеризующий истинный уровень обученности);  $d$  – надежность определения параметра  $p$ ;  $\Phi_0$  – интеграл вероятностей (функция Лапласа);  $\varepsilon$  – точность определения истинного балла обучаемого  $p$ .

Задавая значения величин точности ( $\varepsilon$ ), надежности ( $d$ ) и, учитывая текущий уровень обученности ( $p$ ), с помощью формул можно определить требуемое количество интерактивных слайдов в подготавливаемом курсе обучения. В случае возникновения необходимости проведения комплексной объективной аттестации с высокой степенью точности ( $\varepsilon = 0,05$ ) и надежности ( $d = 0,9$ ), для испытуемых с истинным уровнем обученности  $p = 0,7$  потребуется подготовить  $N = 228$  интерактивных слайдов. Заметим, что объективные зарубежные тесты для профессиональной аттестации (для адвокатов, врачей, бухгалтеров и др.) содержат, как правило, от 220 до 250 тестовых заданий, при  $p$  в диапазоне от 0,7 до 0,75. Однако для определения качества усвоения материалов учебного курса не требуется чрезвычайно высокая точность педагогического измерения. Для этой цели вполне подойдет средняя точность и надежность, поэтому величины  $\varepsilon$  и  $d$  можно положить равными 0,1 и 0,8 соответственно. В этом случае, для определения истинного балла  $p = 0,6$  достаточно подготовить 40 интерактивных слайдов.

Дистанционную (компьютерно-ориентированную) форму обучения логично рассматривать как человеко-машинную систему, в которой обучаемый и преподаватель выступают в роли человека-оператора. Если обучаемый управляет системой приобретения знаний, то преподаватель опосредованно управляет процессом обучения. Как и при проектировании человеко-машинной системы при создании ИСРКОС следует решать вопросы перераспределения функций между человеком и вычислительными средствами.

С этой целью предлагается использовать в процессе обучения индивидуальную модель обучаемого [4]. При этом предполагается, что процесс обучения представляет собой последовательную совокупность

освоения отдельных разделов учебного материала. Каждый этап обучения заканчивается тестированием. Обучаемый может управлять процессом своего обучения посредством выбора способов освоения учебного материала.

Под моделью обучаемого будем понимать нечеткий недетерминированный автомат вида [5]:

$$A = \langle U, X, Y, s_0, \delta, \sigma \rangle, \quad (1)$$

где  $U = \{U_1, U_2, \dots, U_m\}$  – конечное множество входов;

$X = \{X_1, X_2, \dots, X_n\}$  – конечное множество состояний;

$Y = \{Y_1, Y_2, \dots, Y_p\}$  – конечное множество выходов;

$\delta: X \times U \times X \rightarrow [0, 1]$  – функция переходов;

$\sigma: X \times Y \rightarrow L$  – функция выходов;

$s_0$  – начальное состояние.

Функция  $\delta$  порождает множество нечетких матриц перехода:

$$T = \{ \delta_{X_i, X_j}(U), 1 \leq i \leq j \leq n \}, \quad (2)$$

функция порождает нечеткую матрицу выхода. Среди множества состояний автомата выделяется множество финальных (заклочительных) состояний  $X_n$ . Нас будет интересовать такой тип автомата, для которого каждое состояние  $X_i, i \in I = \{1, \dots, n\}$  зависит от предыдущего состояния  $X_{i-1}$ . Подобная зависимость может определяться последовательностью реализации подцелей, приоритетом выполнения и т.п.

В этом случае автомат можно задать как нечеткий граф:

$$G = \{ \mu_G(X_{i-1}, X_i) \in M \}, \quad (3)$$

где  $M$  – множество принадлежностей элементов  $X_{i-1} \times X$ .

Цель обучения декомпозируется на  $i$  последовательных (по времени освоения материала) подзадач. Будем интерпретировать  $X_i$  как множество результатов  $i$ -го теста,  $Y_j, j \in J = \{1, \dots, p\}$  – как множество интервалов времени на обучение,  $L$  – как множество доходов, связанных с реализацией выбранного способа обучения (освоения материала)  $u \in U$  на интервале времени  $Y_j$ . Очевидно, что в рассматриваемом типе автомата  $m = n$  и может трактоваться как число последовательных этапов (шагов) достижения цели.

Управляющие решения и состояния подзадач, протекающие во времени, будем трактовать как нечеткие события на интервале  $Y_k, 1 \leq k \leq p$ . При таком подходе функция переходов может задаваться экспертным путем и отражать уже имеющийся опыт обучения, исходя из практического опыта подачи учебного материала. При этом, естественно, не учитывается результат обучения в зависимости от времени его реализации, а также личностные особенности приобретения знаний конкретным индивидом. Для учета данного обстоятельства необходимо построить индивидуализированную функцию переходов. С этой целью используется исходная информация от

обучаемого, которая включает прогноз применения того или иного способа освоения материала в зависимости от возможных результатов тестирования в виде функции  $\mu: X_i \times U_1 \rightarrow [0, 1]$ , а также прогноз перехода управляемого процесса обучения из исходного состояния  $S_0$  на первом шаге решения в зависимости от ограничений на имеющиеся ресурсы. На основе этой информации программируется автоматная модель. Для этого на каждом шаге решается система композиционных уравнений вида [6]:

$$\left. \begin{aligned} m(X_i)/U_1 &= m(X_{i-1})/U_{i-1} \circ d(X_{i-1}, X_i)/U_1, \\ m(U_1) &= m(X_{i-1}) \circ m(X_{i-1}, X_i)/U_1 \end{aligned} \right\}, \quad (4)$$

где "o" – знак операции "композиция";

$\mu(X_i)/U_1, \mu(X_{i-1})/U_{i-1}$  – нечеткие оценки возможности управляемого процесса находится в состояниях  $X_i, X_{i-1}$ , при применении способов освоения учебного материала  $U_1$  и  $U_{i-1}$ , соответственно;

$\mu(U_1)$  – нечеткая оценка выбора обучаемым способа освоения материала  $U_1$ .

Полученные оценки группируются попарно, исходя из следующего условия:

$$\mu(U_1) \leq \mu(X_i)/U_1. \quad (5)$$

Формирование пар по такому принципу согласуется с реальным выбором решения: результату теста с максимальной оценкой возможности должен соответствовать способ освоения учебного материала также с максимальной оценкой применения его обучаемым. Выделение пар позволяет выявить наиболее возможные связи по способам освоения материала между результатами тестирования каждого этапа обучения. При этом каждый способ освоения, маркирующий связь, характеризуется нечеткой оценкой использования его обучаемым и нечеткой оценкой цены обучения (нормированный доход), субъективно осознаваемой обучаемым, в зависимости, например, от времени или сложности освоения учебного материала.

Построение автомата осуществляется следующим образом [7]. Из исходного состояния в состояния первого шага принятия решений проводятся дуги, маркированные теми способами освоения учебного материала, использование которых, в соответствии с прогнозом, позволяет добиться результатов промежуточного тестирования на первом этапе обучения, характеризующихся наибольшей оценкой. В зависимости от означивания этих дуг и на основе сформированных пар "способ освоения материала" – "результаты тестирования" проводятся означенные дуги от первого до  $n$ -го этапа обучения.

В результате построений получаем нечеткий недетерминированный автомат, моделирующий поведение обучаемого при различных результатах промежуточного тестирования. Применяя подход, используемый в динамическом программировании, можно выделить классы стратегий обучаемого. В первую очередь нас будет интересовать тот класс, стратегии которого позволяют достичь цели обучения и характеризуются максимальными оценками связей между результатами тестов. Для выделения таких стратегий на множестве финальных результатов тестирования определяются результаты, соответствующие цели обучения. Далее выделяются результаты тестирования на (n-1)-м шаге, переход из которых в целевые состояния n-го шага характеризуется способами освоения учебного материала с оценкой, равной

$$a(U_{n-1}) = \max_{Y_{n-1}} (\min(mU_{n-1}), s_{X_{n-1}, U_{n-1}}). \quad (6)$$

Подобная процедура осуществляется для каждого шага решения, вплоть до состояния  $S_0$ . Использование процедуры позволяет выделить возможные стратегии обучения, представляющие собой взвешенные пути на графе от вершины  $S_0$  до вершин из множества  $X_n$ . Каждый q-й путь представляет собой взвешенную относительно способов освоения учебного материала последовательность вида:

$$S_q = (s_0, U_0, X_1^r, U_1, \dots, X_{n-1}^r, U_{n-1}, X_n^r), \quad (7)$$

где  $r_w, w \in \{1, 2, \dots, n\}$  – число результатов w-го теста. Элементы (результаты итогового тестирования) множества  $X_n$  могут представлять для обучаемого различную ценность, что отражается заданием на  $X_n$  нечеткой цели с функцией принадлежности:

$$m(g) = \{m(X_n^1, \dots, X_n^h)\}, \quad (8)$$

где  $h$  – размерность множества  $X_n$ ;

$m(X_n^p) \pi \in \{1, 2, \dots, h\}$  – функция принадлежности результата тестирования нечеткой цели  $g$ .

В этом случае каждую стратегию из класса  $\pi$  можно оценить следующим образом:

$$b(S_q^p) = \min(a(U_0), a(U_1), \dots, a(U_{n-1}), m(X_n^p)). \quad (9)$$

Очевидно, что стратегия, имеющая оценку

$$\max_q (b(S_q^p)), \quad (10)$$

наиболее соответствует индивидуальному стилю приобретения знаний обучаемым, но не всегда может соответствовать максимальной функции принадлежности результата тестирования относительно цели  $g$ . Полученного противоречия при построении модели обучения можно избежать, используя следующие способы:

- изменение предпочтений обучаемого;
- изменение исходных матриц переходов и выходов;
- использование такой организации сопровождения обучения, которая бы учитывала только конечную цель, т.е. состояние нечеткой цели с максимальной функцией принадлежности.

В последнем случае следует использовать смешанные стратегии, образованные частями стратегий из классов  $\pi$ . Необходимость в формировании смешанной стратегии определяет трансформацию стратегии из некоторого класса, выбранную, исходя из критерия (3.9) в стратегию из другого класса, исходя из следующего критерия:

$$\max_p (m(X_n^p)). \quad (11)$$

Очевидно, что необходимость в трансформации возникает в тех случаях, когда выбранная исходная стратегия перестает быть эффективной по отношению к нечетко заданной цели [8]. При этом важно определить те пространственно-временные точки графа (вершины), где подобная трансформация возможна. С этой целью введем понятие коэффициента свободы выбора обучаемого (КСВ), под которым будем понимать отношение числа допустимых стратегий из класса

$$S_q^p \text{ c } b(S_q^p) \geq \epsilon, \quad (12)$$

к общему числу стратегий из этого же класса. Здесь  $\epsilon$  означает порог возможностей обучаемого в достижении поставленной им цели, при использовании индивидуального стиля освоения учебного материала. Чем меньше значение КСВ, тем меньше у обучаемого вариантов в достижении поставленной цели без изменения индивидуальной стратегии. Поэтому обучающая система, построенная на основе модели поведения обучаемого, должна:

- рекомендовать после получения каждого результата тестирования те способы освоения материала, которые являются составными частями допустимых стратегий и не делают значение КСВ равным или близким нулю;

- в случае невозможности выполнения предыдущего пункта трансформировать индивидуальную стратегию обучаемого, переходя на смешанную стратегию обучения. Такая стратегия обучения останется допустимой по отношению к достижению поставленной цели;
- перепрограммировать модель обучения в ситуациях перехода от одного результата тестирования к другому в пределах одного и того же этапа обучения.

В последнем случае необходимость в перепрограммировании модели обучения определяется изменением функции доходов. Использование рассмотренной модели позволяет произвести автоматическую коррекцию индивидуальных стратегий обучаемого. Особенность предлагаемого подхода состоит в возможности перенесения модели обучаемого, сформированной преподавателем при личных контактах, в образовательную среду компьютерного (дистанционного) обучения.

**Основные результаты и выводы.** Предложенный подход позволяет одновременно использовать инструментальную систему в процессе совместной работы над одним учебным курсом коллективу таких специалистов, как преподаватель, ведущий аналогичный курс, психолог, дизайнер, редактор и т.д. Использование инструментальной системы поддержки интеграции учебно-методических материалов позволит не только существенно сократить время создания курсов компьютерного (дистанционного) обучения, но и в дальнейшем позволит оперативно обновлять материалы курсов. Внедрение таких систем позволит обеспечить более быстрый переход на единые форматы передачи учебной информации, что, в свою очередь, будет способствовать расширению сферы применения данной инструментальной системы.

**Список литературы:** 1. Чернявский А. Ф. Автоматизированные обучающие системы на базе ЭВМ / Под ред. А.Ф. Чернявского. – Минск: Изд-во БГУ им. В.И.Ленина, 1980. – 155 с. 2. Румянцев И. А. Аксиоматическая теория структурно-алгоритмического моделирования процесса обучения. – СПб: РГПУ им. А.И.Герцена, 1991. – 230 с. 3. Werner E. Artificial intelligence and tutoring systems. Computational approaches to the communication of knowledge. – Los Altos: Morgan Kaufmann, 1987. – 486 p. 4. Benjamin S. Bloom Taxonomy of Educational Objectives: The Classification of Educational Goals, by committee of college and university examiners. – New York: D. McKay Co, 1956-1964. 5. Семенов В. В., Шубин И. Ю. Концепция интегрированной среды разработчика компьютерных обучающих систем // Проблемы бионики. – 1999. – № 50. – С. 49-61. 6. Bilous N., Shubin I., Vyrodov O. The conception of interactive training system design: In journal of New Solutions in Modern Technologies. – Kharkiv, Ukraine. – 2006. – № 79. – С. 68-70. 7. Bondarenko M., Bilous N., Shubin I. The Ukrainian e-Learning Region: In Proceedings of 10-th International LInE Conference New Partnerships and Lifelong Learning. – Helsinki, Finland. – 2008. – P. 88-92 8. Горбач Т. В., Шубин И. Ю. Концепция построения интерактивной системы реализации дополнительного профессионального образовательного процесса // Материалы 5-й конференции "Дополнительное профессиональное образование: от спроса до признания". – Москва, 7-8 июня 2007 г. – С. 51-53.

Поступила в редколлегию 21.12.09

УДК 519.7

*М. Л. ЛЮБЧИК*, аспирант НТУ «ХП»

### **АНАЛІЗ ПОКАЗНИКІВ ЯКОСТІ ФУНКЦІОНУВАННЯ СИСТЕМИ «БОНУС-МАЛУС» НА ОСНОВІ МАРКІВСЬКИХ МОДЕЛЕЙ**

В статті розглядається задача аналізу системи «бонус-малус» в автомобільному страхуванні. Застосуванням негативної біноміальної моделі для розподілу числа страхових випадків проведено і досліджене моделювання зміни середнього рівня премій, величини нарахування в перший рік і еластичності системи «бонус-малус» для різних видів транспорту.

В статье рассматривается задача анализа системы «бонус-малус» в автомобильном страховании. На базе отрицательной биномиальной модели для распределения числа страховых случаев проведено и исследовано моделирование изменения среднего уровня премий, величины начисления в первый год и эластичности системы «бонус-малус» для различных видов транспорта.

In this paper inverse the problem of analysis of "bonus-malus" in automobile insurance. On the basis of the negative binomial model for the distribution of the number of insurance cases performed and studied modeling changes in the average level of premiums, the value of assessment in the first year and the elasticity of the «bonus-malus» system for different modes of transport.

**Вступ.** Важливим інструментом, що застосовується при страхуванні громадянської відповідальності власників транспортних засобів є так звана система «бонус-малус», що використовується страховими компаніями при розрахунку вартості договору страхування. Система «бонус-малус» (СБМ) передбачає зменшення або збільшення страхової премії стосовно базового страхового тарифу, якій власник транспортного засобу зобов'язаний заплатити при укладанні договору страхування [1]. Коефіцієнт, що знижує страхову премію (бонус), застосовується у випадку, якщо водій не здійснивав шляхово-транспортні випадки (ШТП) в період дії попереднього договору страхування, інакше застосовується коефіцієнт, що підвищує премію (малус)

Метою даної роботи є аналіз показників якості функціонування СБМ на основі статистики страхових випадків, зареєстрованих страховою компанією.

**Постановка задачі.** В основу моделювання СБМ покладене розподіл полісів на кінцеве число класів, що позначені через  $C_i$  ( $i=1...s$ ), так, щоб розмір річної премії залежав тільки від номера класу. Клас, до якого відноситься поліс у поточний період страхування, визначається класом, у якому він перебував у попередній період і числом страхових випадків, що було зареєстровано у даний період. Така система визначається трьома елементами: преміальною шкалою  $b = (b_1 \mathbf{K} b_n)$ , початковим класом  $C_0$  і перехідними правилами, які визначають умови переходу з одного класу в інший, за умови, що число страхових випадків відомо. Ці правила можна

ввести у вигляді перетворень  $T_k$  таких, що  $T_k(i) = j$ , якщо поліс переходить із класу  $C_i$  в клас  $C_j$ , за умови, що зареєстровано  $k$  страхових випадків.

Перетворення  $T_k$  можна представити у вигляді матриці  $T_k = (t_{ij}(k))$ , де  $t_{ij}(k) = 1$ , якщо  $T_k(i) = j$  й  $t_{ij}(k) = 0$  у протилежному випадку.

Ймовірність переходу із класу  $C_i$  в клас  $C_j$  для страховальника характеризується параметром  $\lambda$  - частотою страхових випадків, і має вигляд

$$P_{ij}(\lambda) = \sum_{k=0}^{\infty} P_k(\lambda) t_{ij}(k), \quad (1)$$

де  $P_k(\lambda)$  - ймовірність того, що з вини водія відбудеться  $k$  страхових випадків протягом року. Очевидно, що  $P_{ij}(\lambda) \geq 0$  й  $\sum_{j=0}^S P_{ij}(\lambda) = 1$ . Тоді послідовність переходів водія із класу в клас визначається ланцюгом Маркова з перехідною матрицею

$$P(\lambda) = \|P_{ij}(\lambda)\|. \quad (2)$$

де  $P_{ij}(\lambda)$  - ймовірність переходу водія із класу  $i$  в клас  $j$ .

Найпоширенішими моделями для розподілу числа страхових випадків у страховому портфелі є пуасонівська, негативна біноміальна, зворотна гаусівська модель і модель «гарні/погані ризики» [1]. Пуасонівський розподіл дає гарний опис поведінки індивідуальних страховальників і використовується для порівняння різних СБМ із погляду страховальника, але найбільш точним виявляється негативний біноміальний розподіл [1].

**Вибір моделі розподілу страхових випадків.** Найбільш поширеною є пуасонівська модель, яка передбачує, що розподіл  $\{p_k(\lambda), k = 0, 1, 2, \dots, K\}$  числа страхових випадків відповідає пуасонівському закону:

$$p_k = \frac{e^{-\lambda} \lambda^k}{k!}, \quad k = 0, 1, 2, \dots, K \quad (3)$$

Проста пуасонівська модель не завжди відповідає даним реальних статистичних спостережень. Відхилення гіпотези про пуасонівський вигляд моделі є наслідком того, що поведінка страховальників є неоднорідним. Більш реалістичною є так звана негативна біноміальна модель, що дозволяє задовільно моделювати розподіл числа страхових випадків.

Для отримання негативної біноміальної моделі припустимо, що розподіл  $\{p_k(\lambda), k = 0, 1, 2, \dots, K\}$  числа страхових випадків на рахунку кожного

страховальника відповідає пуасонівському розподілу, у якому значення параметра  $\lambda$  міняється від одного страховальника до іншого. Кожен страховальник характеризується значенням свого параметра  $\lambda$ . При такому підході  $\lambda$  розглядається як випадкова величина  $\Lambda$  з гама-розподілом [2]. Тоді розподіл числа страхових випадків має назву змішаного пуасонівського розподілу і визначається формулою:

$$p_k = \int_0^{\infty} \frac{e^{-\lambda} \lambda^k}{k!} u(\lambda) d\lambda, \quad k = 0, 1, 2, \dots, K \quad (4)$$

де  $u(\lambda)$  - так звана структурна функція, а саме щільність розподілу випадкової величини  $\Lambda$ . Виберемо розподіл випадкової величини  $\Lambda$  у вигляді гамма-розподілу  $u(\lambda) = \frac{\tau^\alpha e^{-\tau\lambda} \lambda^{\alpha-1}}{\Gamma(\alpha)}$ ,  $\alpha, \tau > 0$  з параметрами  $\alpha$  й  $\tau$ .

Негативний біноміальний розподіл сходиться до пуасонівського при  $\alpha \rightarrow \infty$  й при  $\tau \rightarrow 0$  за умови, що середнє залишається постійним. Таким чином, параметр  $\alpha$  може інтерпретуватися як міра концентрації в реалізації страхових випадків із часом. Використання негативного біноміального розподілу замість пуасонівського є розумним, якщо концентрація слабо впливає, тобто якщо  $\alpha$  мало.

Негативна біноміальна модель має дуже важливу теоретичну властивість стабільності структурної функції [3]. Покажемо, що якщо апіорним розподілом випадкової величини  $\Lambda$  є гамма-розподіл з параметрами  $\alpha$  й  $\tau$ , той апостеріорний розподіл також є гамма-розподілом, але з параметрами  $\tau + t$  та  $\alpha + k$ , де  $k = \sum_{i=1}^t k_i$  - загальне число страхових випадків. Таким чином, якщо відбуваються  $k$  аварій у  $t$  років, то це приведе тільки до зміни параметрів гамма-розподілу, а саме,  $\alpha$  і  $\tau$  заміняться  $\alpha + k$  й  $\tau + t$ .

Дійсно, за теоремою Байєса

$$u(\lambda | k_1, \dots, k_t) = \frac{(\tau + t)^{\alpha + k} \lambda^{\alpha + k - 1} e^{-(\tau + t)\lambda}}{\Gamma(\alpha + k)}, \quad (6)$$

що є функцією щільності розподілу-гамма-розподілу з параметрами  $\tau + t$  й  $\alpha + k$  [4].

**Аналіз системи «бонус-малус».** Відповідно до основного рівняння динаміки марківського ланцюга

$$P^T(t+1) = P^T(t) \cdot P, \quad (7)$$

де  $P_{ij} = P(S(t+1) = j | S(t) = i)$  - перехідна матриця,  $P(t) = (P_1(t) \mathbf{K} P_N(t))$  - розподіл ймовірностей знаходження страхувальника у відповідному класі СБМ, що задовольняє умовам нормування ( $\sum_{i=0}^N P_i(t) = 1$ ,  $0 \leq P_i(t) \leq 1$ ),  $t$  - номер року,  $N$  - кількість класів у СБМ.

Використовуючи вектор ймовірностей знаходження страхувальника у відповідному класі СБМ  $P(t)$  у рік  $t$  можна знайти середню премію, що виплачує страхувальник

$$A = (\sum_{i=0}^N P_i(t) * S_i) / N. \quad (8)$$

Знайдений розподіл ймовірностей стану страхувальника дозволяє провести аналіз ефективності СБМ шляхом обчислення показників ефективності, а саме відносного стаціонарного середнього рівня премій  $\rho$ , величини нарахування в перший, вступний рік  $r$  та еластичності середньої стаціонарної премії  $\eta(\lambda)$  щодо частоти страхових випадків  $\lambda$ :

$$\rho = \frac{A - A_{\min}}{A_{\max} - A_{\min}}, \quad r = \frac{A_{\text{in}} - A}{A}, \quad \eta(\lambda) = \frac{dP(\lambda)/P(\lambda)}{d\lambda/\lambda}, \quad (9)$$

де  $A$  - середня премія,  $A_{\min}$  - мінімальна премія,  $A_{\max}$  - максимальна премія,  $A_{\text{in}}$  - початкова премія.

За допомогою розробленої математичної моделі був проведений чисельний аналіз ефективності СБМ, а саме, моделювання зміни середнього рівня премій, величини нарахування в перший рік й еластичності СБМ для різних видів транспорту залежно від часу. При цьому для побудови моделі було використане реальні статистичні дані, отримані від страхових фірм України за 2005-2007 роки.

**Висновки.** З застосуванням отриманих результатів була сконструйована оптимальна по прибутку СБМ, що мінімізує ризик страхової фірми для кожного виду транспорту. Дослідження залежності доходу страхової фірми від значення середнього розміру виплати по страховому випадку для різних видів транспорту дозволило виділити його збиткові види. Математична модель, що розглядається в даній роботі, може бути застосована на практиці для удосконалення систем автомобільного страхування.

**Список литературы:** 1. Лемер Ж. Системы «бонус-малус» в автомобильном страховании: Перев. с англ., изд 2-е. – М.: Янус-К, 2003. – 259с. 2. Лемер Ж. Автомобильное страхование. Актуарные модели: Перев. с англ., изд 2-е – М.: Янус-К, 2003. – 308с 3. Фалин Г.И. Математический анализ рисков в страховании. – М.: Росс. Юр. Изд. Дом, 1994. 4. R.Kaas, M.Goovaerts, J. Dhaene, M.Denuit «Modern Actuarial Risk Theory». – Kluwer Academic Publishers, 2001. – 309 p. – ISBN 0-7923-7636-6

Поступила в редколлегию 14.12.09

УДК 519.2

**Н. К. СТРАТИЄНКО**, канд. техн. наук, доцент каф. АСУ НТУ «ХП»,  
**К. С. СУМЦОВА**, студент каф. АСУ НТУ «ХП»

### ПІДХІД ДО РОЗРОБКИ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ПІДБОРУ ПЕРСОНАЛУ В РЕКРУТІНГОВИХ КОМПАНІЯХ

В статті розглядаються питання автоматизації процесу підбору персоналу в рекрутингових компаніях, наводиться постановка задачі, розглядається підхід до розробки інформаційного забезпечення з використанням методів багатокритеріальної оптимізації для відбору кандидатів на вакантні посади у різні компанії.

В статье рассматриваются вопросы автоматизации процесса подбора персонала в рекрутинговые компании, приводится постановка задачи, рассматривается подход к разработке информационного обеспечения с использованием методов многокритериальной оптимизации для отбора кандидатов на вакантные должности в различные компании.

In the article the questions of automating the process of recruiting staff in the recruiting companies, the target settings is given, although reviews an approach to develop the information, using the methods of multi-criteria optimization for selecting candidates for vacant positions in various companies.

**Вступ.** Вдосконалення систем управління персоналом потребує нові підходи до організації підбору персоналу, а також впровадження інформаційних засобів для обліку, аналізу та підбору працівників.

Рекрутингова компанія (РК), виконуючи своє поставлене завдання, є тією ланкою, яка допомагає швидко і якісно надати кандидатів на вакантну посаду. Для виконання пошуку і відбору кандидатів на різні посади, РК необхідна більш ефективна автоматизація процесу обліку і аналізу даних кандидатів, бо з кожним роком об'єми інформації збільшуються, і процес аналізу стає важким, менш професійним та займає багато часу.

Формування ефективної системи для аналізу даних та відбору кандидатів є одним з найбільш важливих завдань сучасної системи управління персоналом (УП).

**Постановка задачі.** Основною діяльністю рекрутингової компанії є швидкий та якісний підбір персоналу для різних підприємств, організацій, фірм, що працюють в різноманітних сферах діяльності. РК мають як переваги так і недоліки перед відділами кадрів підприємств.

При скоординованій роботі підприємства і РК, можна досягти ефективного пошуку нових кадрів для підприємства. Це означає формування чітких вимог до вакансії, а також тісну скоординовану роботу співробітників відділу кадрів і консультанта РК, який працює над конкретною вакансією.

Внутрішня схема взаємодії РК і компанії-замовника приведена на рис. 1.



Рис. 1- Структура схеми взаємодії РК і компанії-замовника

Як зазначають автори: [1] - [3], для покращення якості процесу пошуку нового персоналу РК потребується рішення сукупності завдань. До них відносяться: спрощення, прискорення та вдосконалення процесу обліку великих об'ємів даних кандидатів; створення чіткої структури даних для пошуку потрібної інформації; прискорення процесу подання резюме кандидатами на відповідні вакансії та інше.

Тобто виникає потреба в розробці інформаційного та програмного рішення для автоматизації процесу відбору персоналу, на основі використання алгоритмів знаходження потрібних кандидатів, керуючись висунутими вимогами до вакансії з використанням багатьох параметрів для пошуку.

Для вирішення поставленої задачі, опираючись на вивчення та аналіз предметної області, необхідно розробити програмне забезпечення, яке відповідатиме наступним вимогам, а саме:

- забезпечувати комфортний доступ до бази даних резюме;
- підтримувати одночасну роботу користувачів;

- надавати швидкий доступ до потрібної інформації
- мати просту та ефективну систему безпеки;
- бути надійним, та переносним;
- мати здібність вміщування, зберігання та обробки великих об'ємів даних;
- надавати можливість формування списку потрібних кандидатів по вимогам компанії, керуючись ідеальним профілем кандидата (що складається з вимог компанії-замовника), критеріями для відбору виступають особові якості та професіоналізм кандидата (лідерство, організаційні навички, професіоналізм та інше, які формалізуються), альтернативами є кандидати на посаду, ваги критеріїв – ступінь важливості того чи іншого критерію для конкретної вакансії, а в якості особи що приймає рішення виступає експерт, тобто консультант рекрутингової компанії;
- забезпечувати можливість розміщення інформації о вакансіях у Internet на сайті РК, огляду інформації щодо консультантів РК, заповнення форми для відправки резюме, огляду компанії з якими співпрацює РК, доступу до пакетів з прикладами документів для складання договору співпраці, умов співпраці та інше.

**Інформаційне забезпечення для вирішення задачі.** В даній статті пропонується підхід до розробки інформаційного забезпечення для вирішення поставленої задачі, яке здатне забезпечити можливість чіткого аналізу даних кандидатів, проведення тестування претендентів, проведення пошуку претендентів на вакансії на базі визначених критеріїв, проведення оцінки їх компетенцій шляхом використання сучасних методик оцінки персоналу (інтерв'ю за компетенціями, оцінка лідерських якостей, проведення тесту на IQ та інше) за допомогою вирішення багатокритеріальної задачі. Також в статті відзначено спрямованість на спеціалізацію діяльності рекрутингових компаній, та можливість використання інформаційного забезпечення відділами з рекрутингу в рамках кадрової організації підприємств.

У процесі проектування інформаційної системи (ІС) побудовано діаграму варіантів використання, яка зображена на рис. 2.

Після збору та аналізу вимог була побудована система бізнес-правил, та на її основі побудована концептуальна модель даних ІС, для подальшого проектування бази даних.

Система бізнес-правил, згідно досліджуваної предметної області:

- “Кандидат” – особа, що знаходиться в пошуку робочого місця в даний час, або зацікавлена в розгляді відповідних вакансій, з метою можливості отримання бажаного робочого місця в майбутньому часі.
- “Кандидат” може: мати декілька робочих місць (посада і досвід роботи на яких можуть бути різними); мати водійські права (А, В, С,

D, E, при цьому можуть бути права різної категорії, наприклад «А, В»); проживати в 1 місті; брати участь в декількох тренінгах або конференціях; здобути декілька вищих освіт і відповідно їх типів; володіти декількома іноземними мовами; мати декілька контактних телефонних номерів.

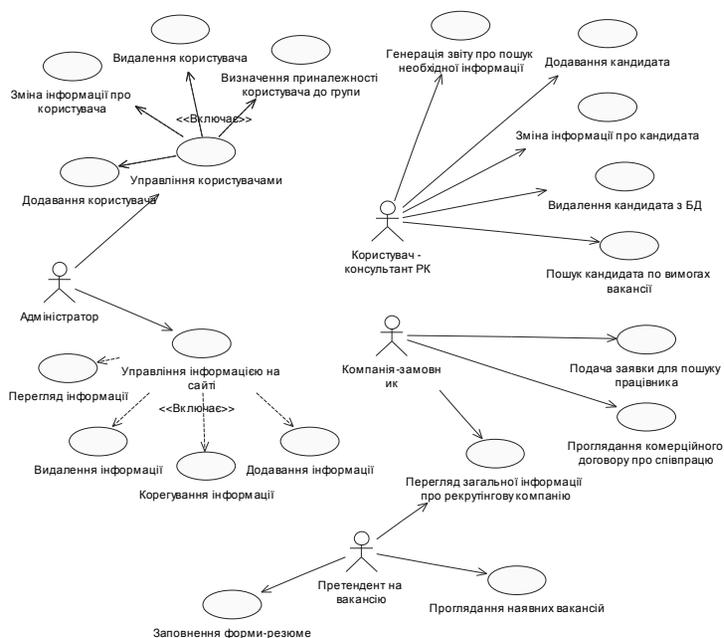


Рис. 2. Діаграма варіантів використання ІС

- “Тип освіти” ділиться на такі напрями, як: технічне, філологічне, економічне, юридичне, медичне (стоматологічне, гінекологічне, хірургічне, і ін.), психологічне, фармацевтичне, хімічне (органічна хімія, неорганічна хімія), і так далі.
- “Компанія” – це компанія (компанія-замовник), підприємство або будь яка організація, зацікавлена в пошуку працівника на вільну вакансію, представляє роботодавця.
- “Компанія”, працює в одній сфері (виробництво, торгівля, будівництво, і так далі), може мати декілька вільних вакансій.

- “Замовлення компанії” має: назву посади, ряд вимог до працівника (вік, тип вищої освіти, наявність водійських прав, мінімальний досвід роботи, кількість підлеглих на колишній роботі, додаткові коментарі до вимог), дату заявки замовлення підбору працівника на необхідну посаду і остаточну дату роботи консультанта РК над вакансією.
- “Консультант РК” – працівник РК, особа, що здійснює підбір персоналу в різні компанії, які можуть працювати в різних виробничих або не виробничих сферах.
- “Консультант РК” має одну посаду і може працювати по декількох вакансіях компаній-замовників.

Для вирішення задачі підбору кандидатів була запропонована математична модель. Задача підбору персоналу вирішується з використанням методів багатокритеріальної оптимізації.

При формуванні підбору персоналу використовуються алгоритми, на основі методів багатокритеріальної оптимізації, які дозволять надати структуровану інформацію, засновану на виділенні певних критеріїв для прийняття рішення консультантом, враховуючи особливості даної наочної області.

Особливостями програмного рішення є:

- його web-орієнтованість,
- адаптованість до особливостей діяльності РК,
- можливість обрання кандидатів з сформованого списку, після введення вимог до кандидата,
- підтримка необмеженої кількості користувачів, одночасно працюючих з базами даних,
- можливість створення звітів та різних документів, з шаблонів,
- можливість використання відділами з рекрутингу в рамках кадрової організації підприємств.

**Висновки.** Для продуктивної роботи сучасні організації практикують підбір персоналу, співпрацюючи з кадровими чи рекрутинговими агентствами.

Запропонований в статті підхід до розробки інформаційного забезпечення, заснований на використанні математичних моделей та комплексній методиці оцінки персоналу, надає можливість підвищити ефективність підбору кадрів.

Подальша робота буде присвячена впровадженню інформаційного та програмного забезпечення, що розроблено, в рекрутингову компанію.

**Список літератури:** 1. Грачєв М. В. Суперкадры: Управление персоналом в международной корпорации.– М.: Дело ЛТД, 2006.– 302 с. 2. Иванова С. В. Искусство подбора персонала: Как оценить человека за час.– М.: Альпина Бизнес Букс, 2004.– 160 с. 3. Сливак В. А. Организационное поведение и управление персоналом.– СПб: Питер, 2000.– 416 с.

**В. В. КОНДРАЩЕНКО**, аспирант НТУ «ХПИ»,  
**В. В. МОСКАЛЕНКО**, канд. техн. наук,  
**Т. В. ЗАХАРОВА**, аспирант НТУ «ХПИ»

### **АРХИТЕКТУРА СППР ДЛЯ ПОСТРОЕНИЯ СХЕМЫ ФИНАНСИРОВАНИЯ ИНВЕСТИЦИОННОГО ПРОЕКТА**

Пропонується загальна схема розв'язання задачі вибору оптимальної форми та джерел фінансування інвестиційного проекту. Розглянуто основні модулі системи підтримки прийняття рішень для побудови схем фінансування інвестиційних проектів із залученням різноманітних джерел. Представлено діаграму компонентів СППР.

Предлагается общая схема решения задачи выбора оптимальной формы и источников финансирования инвестиционного проекта. Рассмотрены основные модули системы поддержки принятия решений по построению схем финансирования инвестиционных проектов с привлечением различных источников. Представлена диаграмма компонентов СППР.

The general scheme of solving the task of choosing the optimal form and source of financing the investment projects is proposed. The main modules of decision support system for computation of the financing schemes for investment projects with different sources are considered. The components diagram of decision support system is presented.

**Введение.** Накопленный человечеством опыт в организации и управлении различными проектами получил свое отражение в современной теории «управления проектами». Среди прочих основных задач управления проектами, отдельно следует рассматривать задачу финансирования проекта. Финансирование проекта в общем случае должно обеспечить решение двух основных задач: обеспечить такую динамику инвестиций, которая бы сделала возможным использование проекта в соответствии с временными и денежными ограничениями; уменьшить проектные риски за счет оптимизации структуры финансирования и максимизации налоговых льгот. Так как данная задача относится к задачам управления проектами, то и информационные технологии для управления процессами финансирования проектов зачастую являются подсистемами более крупных продуктов, охватывающих другие аспекты планирования и управления проектами [1]. Обзор информационных технологий в данной области показывает недостаточно представленную функциональность для разработки и анализа схем финансирования инвестиционных проектов. В данной работе рассмотрим разработку программного комплекса, который позволит осуществлять выбор источников привлечения денежных средств и производить расчеты схем финансирования проектов с привлечением сторонних источников таких, как банковский кредит, прямые инвестиции и привлечение денежных средств путем продажи акций предприятия на фондовом рынке [2-4].

**Общая схема решения задачи выбора оптимальной формы и источника финансирования.** Задача построения схемы финансирования, как подзадача управления проектом, состоит из следующего ряда шагов:

- подготовка необходимых данных об инвестиционном проекте;
- анализ и подготовка данных о ситуации на финансовом рынке в плановом периоде;
- анализ потенциальных источников стороннего финансирования;
- подготовка необходимых данных об источниках привлечения денежных средств, которые предстоит более детально проанализировать;
- выполнение построения схемы финансирования инвестиционного проекта с привлечением каждого из рассматриваемых источников;
- сравнительный анализ полученных схем и принятие решения о выборе и дальнейшей проработке той или иной схемы.

Этап подготовки данных об инвестиционном проекте для решения данной задачи заключается в анализе необходимых работ и ресурсов с целью рационального разбиения общего объема работ на этапы. Каждому этапу будут соответствовать конкретные задачи по проекту, а также агрегированная оценка необходимых ресурсов, приведенная к стоимостному выражению.

Анализ и подготовка данных о ситуации на финансовом рынке позволяют произвести оценку агрегированного дисконтирующего коэффициента, который служит для корректировки денежных потоков в будущих этапах реализации проекта.

Анализ потенциальных источников стороннего финансирования требуется проводить в том случае, если предприятие, занимающееся реализацией проекта, не в состоянии самостоятельно профинансировать проект и существует необходимость в привлечении денежных средств. Из возможных источников в данной работе рассматривается банковское кредитование, прямое инвестирование и инструменты фондового рынка. Таким образом, анализ потенциальных источников привлечения денежных средств будет включать исследование предложений банков по кредитованию инвестиционных проектов, поиск инвесторов, заинтересованных в финансировании данного проекта, и анализ инструментов фондового рынка.

В случае, когда анализ потенциальных источников привлечения денежных средств выявил несколько привлекательных вариантов, необходимо провести сравнительный анализ этих вариантов с точки зрения их экономической эффективности. Оценка экономической эффективности на основании разработанных математических моделей производится совместно с расчетом оптимальной схемы финансирования проекта, поскольку вариант схемы существенно влияет на этот показатель. Расчет схемы выполняется при помощи математической модели, которая соответствует типу финансирования.

В результате исследователь получает поэтапную схему финансирования данного инвестиционного проекта, которой соответствует максимальная экономическая эффективность реализации проекта в плановом периоде. Таким образом, алгоритм решения задачи состоит в следующем (рис. 1):

- осуществляется сбор данных о возможных источниках финансирования, которые группируются по формам: собственные, заемные, привлеченные денежные средства;
- для каждого источника банковского кредитования  $\{A_1...A_n\}$ , и внешних инвестиций  $\{B_1...B_m\}$  выполняется анализ и подготовка данных, необходимых для применения соответствующего инструмента расчета;
- если предприятие является акционерным обществом, то выполняется анализ фондового рынка и подготовка данных, необходимых для применения инструмента планирования реализации акций  $C_1$  (продажа акций на фондовом рынке),  $C_2$  (дополнительная эмиссия акций);
- для каждого источника  $\{A_1...A_n\}$ ,  $\{B_1...B_m\}$  применяется инструмент расчета схемы (в соответствии с формой финансирования) и производится оценка  $\{E_{A_1}...E_{A_n}\}$ ,  $\{E_{B_1}...E_{B_m}\}$ ,  $\{E_{C_1}, E_{C_2}\}$  от реализации проекта на конец рассматриваемого периода; при возможности использования инструментов фондового рынка, аналогичная процедура используется и для способов  $C_1$  и  $C_2$ ;
- производится сравнение полученных значений  $\{E_{A_1}...E_{A_n}\}$ ,  $\{E_{B_1}...E_{B_m}\}$ ,  $\{E_{C_1}, E_{C_2}\}$ , в результате которого выбирается способ  $X \in \{A_1...A_n, B_1...B_m, C_1, C_2\}$ , при котором было достигнуто максимальное  $E \in \{E_{A_1}...E_{A_n}, E_{B_1}...E_{B_m}, E_{C_1}, E_{C_2}\}$ ;
- полученная в результате расчета схема финансирования  $X$  считается наилучшей для данного инвестиционного проекта.

#### Архитектура системы поддержки принятия решений.

Автоматизированные системы управления проектами в большинстве случаев представляют собой пакеты календарного планирования, которые могут быть разделены на профессиональные и настольные системы. Профессиональные системы предоставляют более гибкие средства реализации функций планирования и контроля, но требуют больших затрат времени на подготовку и анализ данных и, соответственно, высокой квалификации пользователей. От пользователей, использующих пакеты планирования лишь время от времени при необходимости спланировать небольшой комплекс работ или ввести фактические данные по проекту, трудно ожидать серьезных затрат времени и усилий на то, чтобы освоить и держать в памяти какие-либо специфические

функции планирования или оптимизации расписаний. Для них более важным является простота использования и скорость получения результата.



Рис. 1. Общая схема выбора оптимального источника финансирования

Рассмотрим следующие средства, которые реализуют функциональные возможности планирования и управления проектами [5]:

Средства описания комплекса работ проекта, связей между работами и их временных характеристик:

- описания глобальных параметров планирования проекта;
- описание логической структуры комплекса работ;
- многоуровневое представление проекта;
- назначение временных параметров планирования задач;
- поддержка календарей отдельных задач и проекта в целом.

Средства поддержки информации о ресурсах и затратах по проекту и назначения ресурсов и затрат отдельным работам проекта:

- организационная структура исполнителей;
- ведение списка наличных ресурсов, номенклатуры материалов и статей затрат;
- поддержка календарей ресурсов;
- назначение ресурсов по работам;
- календарное планирование при ограниченных ресурсах.

Средства контроля над ходом выполнения проекта:

- фиксация плановых параметров расписания проекта в базе данных;
- ввод фактических показателей состояния задач;
- ввод фактических объемов работ и использования ресурсов;
- сравнение плановых и фактических показателей и прогнозирование хода предстоящих работ.

Графические средства представления структуры проекта, средства создания различных отчетов по проекту:

- диаграмма Ганта (часто совмещенная с электронной таблицей и позволяющая отображать различную дополнительную информацию);
- PERT диаграмма (сетевая диаграмма);
- создание отчетов, необходимых для планирования и контроля.

Наиболее распространенными продуктами в секторе систем управления проектами являются Open Plan Professional (Welcom Corp., США), Primavera Project Planner (Primavera Systems, Inc., США), Microsoft Project (Microsoft Corp., США); кроме этого имеется ряд продуктов европейских, российских и украинских компаний, например, Spider Project (Spider Technologies Group, Россия). Данные системы принадлежат к различным классам систем управления проектами и обладают своими достоинствами и недостатками.

Однако на данный момент в программных комплексах, представленных на рынке IT продуктов, в недостаточном объеме представлена или отсутствует функциональность построения схем финансирования проектов на основе нечетких данных о проекте или источниках внешнего финансирования, построения схем финансирования с привлечением инструментов фондового рынка для привлечения денежных средств; отсутствуют средства автоматизированного построения оптимальной схемы финансирования и инструментов анализа экономической эффективности привлечения инвестиционных средств от того или иного источника финансирования.

В данной работе предлагается программная реализация СППР для решения ряда задач по финансированию проектов. Спецификой данного программного продукта является то, что он может быть использован, как в качестве самостоятельного приложения, так и в качестве встраиваемой подсистемы в более крупные программные решения (например, в комплексные программные продукты для управления проектами).

Так как на современном рынке информационных технологий наблюдается тенденция переноса настольных программных продуктов в Internet, с последующим предоставлением санкционированного доступа к этим продуктам из любой точки мира, то реализацию базового ядра приложения следует делать максимально гибкой, с тем, чтобы без изменений исходного кода его можно было использовать в различном окружении. Максимальной гибкости удастся достичь, если в архитектуре приложения

выделить отдельные модули, которые последовательно расширяют функциональность друг друга.

Данный программный комплекс представим в виде трех уровней. Ядром приложения (низовой уровень) будет выступать модуль, реализующий технологии расчета схем финансирования инвестиционных проектов. На следующем уровне разработан модуль работы с данными, который предоставляет функциональность сохранения и загрузки данных о проектах, инвесторах, банках и прочих артефактах, и предоставляет эти данные расчетному модулю. На самом верхнем уровне находится модуль пользовательского интерфейса, который в удобной для пользователя форме предоставляет всю необходимую функциональность для построения схемы финансирования инвестиционного проекта.

Модуль пользовательского интерфейса может быть реализован как настольное приложение, или как web-приложение, при этом принцип взаимодействия модулей не изменится.

В случае интеграции функциональности построения схем финансирования инвестиционных проектов в другие системы, потребуется дополнительная разработка модуля-посредника, который обеспечит взаимодействие расчетного модуля с основным продуктом. В редких случаях, при интеграции возможно, также, использование модуля данных, однако зачастую такой подход будет неприемлем, и выглядит целесообразным лишь при интеграции с другим web-приложением.

В рамках данной работы предлагается реализация расчетного модуля, модуля данных и модуля пользовательского интерфейса в виде настольного приложения.

В качестве основной технологии программирования предлагается использовать Java. Данная технология позволяет разрабатывать кросс-платформенные приложения, что снимает требование к работе пользователя с определенной операционной системой, а также позволяет интегрироваться в web-приложения, независимо от того какая операционная система используется на web-сервере. Другое важное преимущество Java в наличии широкого набора библиотек, которые позволяют эффективно создавать настольные и web-приложения. Эта особенность имеет большое значение для разработчиков, и, как следствие, современные гибкие продукты в области управления проектами и управления процессами финансирования проектов разрабатываются преимущественно на Java. Последнее обстоятельство также необходимо учитывать, говоря о перспективах интеграции данной функциональности в другие системы.

Архитектура настольной СППР для построения схем финансирования инвестиционных проектов в контексте использования технологии Java приведена на рис. 2.

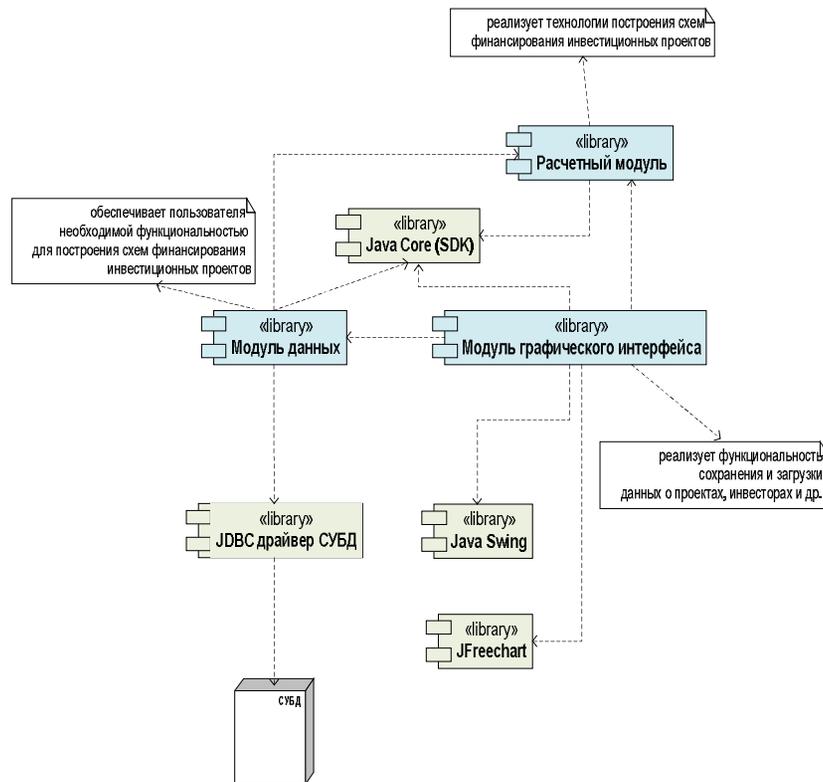


Рис. 2. Диаграмма компонентов СПМР для построения схем финансирования инвестиционных проектов

Расчетный модуль реализуется на основе исключительно базовых библиотек Java SDK, что обеспечивает отсутствие зависимых внешних библиотек и способствует свободной интеграции библиотеки в другие приложения. Функции, реализующие математический расчет, являются частями соответствующих классов.

Модуль данных реализует взаимодействие с базой данных посредством библиотеки JDBC. Следовательно, разрабатываемое настольное приложение может функционировать во взаимодействии с любой СУБД, для которой предоставляется JDBC драйвер. В том числе, JDBC позволяет работать с

данными, которые хранятся на удаленном сервере баз данных, локальном сервере баз данных, могут использоваться встроенные СУБД, XML файлы и прочее. Выбор и использование той или иной СУБД является лишь вопросом конфигурации приложения. В данной реализации настольной СПМР предлагается использовать встроенную СУБД SQLite. Что упрощает установку и перенос программного обеспечения для конечного пользователя, и в то же время обеспечивает всю необходимую функциональность для быстрого и надежного взаимодействия с данными.

Данные, которые необходимо хранить в приложении, определяются в ходе анализа предметной области. Эта информация дает возможность построить концептуальную модель данных, которая будет являться основой для разработки физической базы данных. В данной работе для построения концептуальной модели данных используется методология IDEF1X

Модуль графического интерфейса строится на основе библиотеки Swing, которая входит в поставку Java SDK. Использование этой библиотеки позволяет сделать графический интерфейс пользователя одинаковым для всех поддерживаемых операционных систем, при этом элементы управления являются привычными для пользователя. Также в графическом интерфейсе используется модуль JFreechart, сторонняя бесплатная библиотека, предоставляющая функциональность построения различных графиков на формах.

**Выводы.** Таким образом, в данной работе предложена общая схема решения задачи выбора схемы финансирования проектов, а также архитектура СПМР по финансированию проектов. Разработанная система позволит автоматизировать расчет схем финансирования инвестиционных проектов с привлечением сторонних источников финансирования таких, как кредит, прямые инвестиции и привлечение денежных средств путем продажи акций на фондовом рынке, а также провести анализ предложенных схем.

**Список литературы:** 1. Москаленко В. В., Кондращенко В. В. Применение системного подхода к решению задачи выбора схемы финансирования инвестиционного проекта // Материали XV міжнародної конференції з автоматичного управління (Автоматика-2008), м. Одеса, 23 – 26 вересня 2008р. Ч.1- Одеса: ОМА, 2008. – С. 378-380. 2. Годлевский М. Д., Москаленко В. В., Кондращенко В. В. Система поддержки принятия решений процесса финансирования инвестиционного проекта // Вісник Національного технічного університету «Харківський політехнічний інститут»: Тематичний випуск «Системний аналіз, управління та інформаційні технології». – Харків: НТУ «ХПІ». – 2007. – №5. – С. 75-89. 3. Godlevskiy M., Moskalenko V., Kondrashchenko V. Modeling of the analytical data of investment project financing process // Information systems technology and its applications. International conference ISTA' 2007/ - P.78-90. 4. Годлевский М. Д., Москаленко В. В., Кондращенко В. В. Динамическая модель процесса финансирования инвестиционного проекта // Системний аналіз та інформаційні технології: Матеріали ІХ Міжнародної науково-технічної конференції (15-19 травня 2007р., Київ). – К.:НТУУ «КПІ», 2007. –С. 99. 5. Дубовик М.А Программное обеспечение для управления проектами // [www.projectmanagement.ru](http://www.projectmanagement.ru). – 2007.

**В. В. КОНДРАЩЕНКО**, аспирант НТУ «ХПИ»

### ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СППР ДЛЯ ПОСТРОЕНИЯ СХЕМЫ ФИНАНСИРОВАНИЯ ИНВЕСТИЦИОННОГО ПРОЕКТА

Представлено технологію побудови схеми фінансування інвестиційних проектів з залученням банківського кредиту та функціональні можливості системи підтримки прийняття рішень. Запропонована структура програмного комплексу, описані основні класи. Для роботи з програмним комплексом запропонований інтерфейс користувача, описані основні екранні форми для введення вихідної інформації та для виводу проміжних і підсумкових розрахунків.

Представлена технология построения схем финансирования инвестиционных проектов с привлечением банковского кредита и функциональные возможности системы поддержки принятия решений. Предложена структура программного комплекса, описаны основные классы. Для работы с программным комплексом предложен интерфейс пользователя, описаны основные экранные формы для ввода исходной информации и для вывода промежуточных и итоговых расчетов.

The technology of computation the financing schemes for investment projects with attraction of the bank credit and the main functions of decision support system are presented. The software complex structure is offered. The main classes are described. The user interface of software complex is offered. Main screenshots for input the data and output the intermediate and final results are described.

**Введение.** Задача финансирования проектов относится к задачам управления проектами, поэтому программные комплексы для управления этими процессами зачастую являются подсистемами более крупных продуктов, охватывающих и другие аспекты планирования и управления проектами [1-3]. На данный момент в существующих системах, представленных на рынке программных продуктов, отсутствуют модули построения схем финансирования проектов на основе нечетких данных о проекте; на основе смешанного привлечения денежных средств, т. е. как с участием собственного капитала, так и инвестиций от внешних источников; с привлечением инструментов фондового рынка; отсутствуют средства автоматизированного построения оптимальной схемы финансирования и инструментов анализа экономической эффективности привлечения того или иного источника [4, 5]. Поэтому в данной работе предлагается технология формирования схем за счет привлечения банковских кредитов, а также программный комплекс, реализующий систему поддержки принятия решений (СППР) по построению схем финансирования проектов.

**Технология построения схемы финансирования инвестиционного проекта с привлечением банковского кредита.** Привлечение банковского кредита в качестве источника стороннего финансирования не требует учета финансового интереса стороннего инвестора (банка), как в случае с прямыми инвестициями [4, 5]. Это связано с тем, что коммерческий интерес банка

выражается в параметрах займа, таких как процентная ставка, схема погашения и длительность кредита.

Будем считать, что все ресурсы, необходимые для выполнения проекта, представлены в стоимостном выражении, а весь период выполнения разбит на  $N$  равных этапов. Для каждого этапа считаются известными объем средств  $d_k (k = \overline{1, N})$ , необходимый для вложения (инвестиции, затраты), и чистый доход проекта  $p_k (k = \overline{1, N})$ , который ожидается получить на данном этапе.

Для непрерывной реализации проекта необходимо, чтобы в начале каждого этапа оборотных средств было достаточно для покрытия расходов на этом этапе. Для этого могут быть использованы два источника финансирования проекта: целевые средства, выделенные на реализацию проекта предприятием, и банковский кредит. Целевые средства выделяются однократно в начале первого этапа, а кредит может быть взят в начале любого этапа, кроме последнего (при этом сумма кредита поступает в оборотные средства предприятия без промедления). В данной работе под оборотными средствами будем понимать денежные средства либо их эквиваленты, которые могут быть быстро переведены в денежную форму. Другие виды оборотных средств не рассматриваются. Размер кредита и длительность его использования ограничены. Кредит характеризуется размером, длительностью использования и схемой погашения. Погашение кредита осуществляется поэтапно на протяжении всего срока использования; величина срочной уплаты определяется видом проекта и схемой погашения кредита. Принятие решения на каждом из этапов о взятии кредита, его размере и длительности использования, определяется запасом оборотных средств на предприятии. Обозначим уровень запасов денежных средств в конце  $k$ -го этапа  $s_k$ . Тогда полагая, что начальный уровень запасов оборотных средств равен  $Q$ , можно записать  $s_0 = Q$ . Уровень запасов денежных средств на  $k$ -м этапе равен:

$$s_k = s_{k-1} - d_k + p_k + x_k - \sum_{i=1}^{k-1} Y_i \quad (1)$$

где  $s_{k-1}$  – уровень запасов денежных средств для инвестирования в конце  $(k-1)$ -го этапа;

$d_k$  – затраты на реализацию проекта в  $k$ -м этапе;

$p_k$  – поступления денежных средств от внедрения проекта в  $k$ -м этапе;

$x_k$  – размер кредита, взятого в начале  $k$ -го этапа;

$\sum_{i=1}^{k-1} Y_i$  – суммарные затраты на погашение кредитов, взятых на предыдущих этапах (как суммы погашения основного долга, так и начисленные проценты).

Затраты в  $k$ -м этапе на погашение кредита, взятого в  $i$ -м этапе, определяются следующим образом:

$$Y_i^{t_i} = \begin{cases} Y_i(x_i, n_i), & \text{если } (k-i) \leq n_i \text{ и } k > i, \\ 0, & \text{если } (k-i) > n_i \text{ или } k < i \end{cases} \quad (2)$$

где вид функции  $Y_i$  определяется типом схемы погашения кредита  $t_i$ . Ниже приводятся формулы расчета срочной уплаты  $Y_i$  для рассматриваемых схем погашения кредитов:

1. Погашение долга равными срочными платежами:

$$Y_i^1 = \begin{cases} x_i \cdot \frac{(1+r(n_i))^{n_i}}{1+r(n_i)}, & \text{если } (k-i) \leq n_i \text{ и } k > i, \\ 0, & \text{если } (k-i) > n_i \text{ или } k < i \end{cases} \quad (3)$$

2. Погашение займа равными выплатами основного долга:

$$Y_i^2 = \begin{cases} \left[ x_i - \frac{x_i}{n_i} (i-k-1) \right] \cdot i + \frac{x_i}{n_i}, & \text{если } (k-i) \leq n_i \text{ и } k > i, \\ 0, & \text{если } (k-i) > n_i \text{ или } k < i \end{cases} \quad (4)$$

3. Выплаты изменяются в арифметической прогрессии:

Если  $(k-i) \leq n_i$  и  $k > i$ , то

$$Y_i^3 = \left[ x_i - \left( \frac{x_i}{n_i} - \frac{(n_i-1) \cdot d}{2} \right) \cdot (i-k-1) - d \cdot \sum_{j=1}^{i-k-2} j \right] \cdot r(n_i) + \frac{x_i}{n_i} - \frac{(n_i-1) \cdot d}{2} + (i-k-1) \cdot d, \quad (5)$$

где  $d$  – разность арифметической прогрессии.

Если  $(k-i) > n_i$  или  $k < i$ , то  $Y_i = 0$ .

4. Выплаты изменяются в геометрической прогрессии:

Если  $(k-i) \leq n_i$  и  $k > i$ , то

$$Y_i^4 = \left[ x_i - h_1 \cdot \sum_{j=0}^{i-k-2} q^j \right] \cdot r(n_i) + h_1 \cdot q^{i-k-1}, \quad (6)$$

где  $q$  – знаменатель геометрической прогрессии;

$$h_1 = \begin{cases} x_i \cdot \frac{1-q}{1-q^{n_i}}, & \text{если } q < 1 \\ x_i \cdot \frac{q-1}{q^{n_i}-1}, & \text{если } q \geq 1 \end{cases}.$$

Если  $(k-i) > n_i$  или  $k < i$ , то  $Y_i = 0$ .

Таким образом, в каждом этапе на протяжении периода использования кредита ( $(k-i) \leq n_i$  и  $k > i$ ) должны производиться отчисления на погашение кредита, а после ( $(k-i) > n_i$ ) – отчисления равны нулю. Тогда для каждого  $k$ -го этапа может быть выбран кредит со схемой погашения  $t_k \in \{1, 2, 3, 4\}$ , где цифра 1, 2, 3, 4 означает вышеприведенные способы вычисления выплат погашения кредитов.

Для поддержания уровня оборотных средств, достаточного для покрытия инвестиционных затрат на каждом этапе, необходимо, чтобы выполнялось условие  $s_k \geq 0$ .

Величину процентных выплат по кредиту  $W_k$ , взятому в  $k$ -м этапе, можно рассчитать на основании следующей формулы:

$$W_k = x_k \cdot (1+r(n_k))^{n_k} - x_k,$$

где  $r(n_k)$  – величина процентной ставки по кредиту, приведенная к размерности [% в этап].

Тогда суммарные процентные выплаты будут равны сумме процентных выплат по кредитам, взятым в каждом из этапов:

$$W = \sum_{k=1}^N [x_k \cdot (1+r(n_k))^{n_k} - x_k]$$

где  $x_k = 0$  для этапов, в которых кредит брать не нужно и  $x_k \neq 0$  в противном случае.

Учет ограниченности размера кредита произведем путем введения констант  $x^{\min}$  и  $x^{\max}$ .  $x^{\min}$  определяет минимальный размер кредита, который может быть взят на одном этапе,  $x^{\max}$  – соответственно максимальный

размер кредита. Аналогично длительность использования кредита ограничивается константами  $n^{\min}$  и  $n^{\max}$ .

Таким образом, можно построить следующую математическую модель:

$$W = \sum_{k=1}^N [x_k \cdot (1 + r(n_k))^{n_k} - x_k] \quad (7)$$

$$s_k = s_{k-1} - d_k + p_k + x_k - \sum_{i=1}^{k-1} Y_i^{t_i}, k = \overline{1, N} \quad (8)$$

$$x^{\min} \leq x_k \leq x^{\max}, k = \overline{1, N} \quad (9)$$

$$n^{\min} \leq n_k \leq n^{\max}, k = \overline{1, N} \quad (10)$$

$$t_k \in \{1, 2, 3, 4\}, k = \overline{1, N} \quad (11)$$

$$s_0 = Q \quad (12)$$

$$s_k \geq 0, k = \overline{1, N}, \quad (13)$$

где  $Y_i^{t_i}$  ( $t_i \in \{1, 2, 3, 4\}$ ) определяются по формулам (3)-(6).

Следовательно, необходимо определить такие значения  $x_k$ ,  $n_k$  и выбрать схемы погашения кредитов  $t_k$  ( $k = \overline{1, N}$ ), которые удовлетворяют условиям (8)-(13) и доставляют минимум функции (7).

Ввиду того, что исходная задача имеет высокую размерность  $3N$ , где  $N$  – количество этапов реализации проекта, в первую очередь предлагается выполнить преобразования, направленные на уменьшение размерности модели. Классический подход к достижению данной цели состоит в преобразовании исходной математической модели к задаче динамического программирования. Будем рассматривать  $N$ -шаговый процесс оптимизации, при котором на каждом шаге определяются условно-оптимальные затраты на все взятые кредиты до этого шага включительно. Величина этих затрат равна  $W_k$ :

$$W_k = \min_{x_k, n_k} \{W_{k-1} + (x_k (1 + r(n_k))^{n_k} - x_k)\}$$

Это соотношение показывает, что суммарные начисленные проценты на  $k$ -м этапе определяются как сумма начисленных процентов на  $(k-1)$ -м этапе плюс величина начисленных процентов по кредиту, взятому в  $k$ -м этапе. Соответственно, если в  $k$ -м этапе кредит брать не нужно, то  $x_k$  будет

равно нулю и  $W_k = W_{k-1}$ . На каждом шаге необходимо обеспечить неотрицательный уровень запасов оборотных средств, достаточный для покрытия затрат на текущем этапе. Уровень запасов оборотных средств  $s_k$  определяется по формуле:

$$s_k = s_{k-1} - d_k + p_k + x_k - \sum_{i=1}^{k-1} Y_i.$$

В результате на основании задачи нелинейного программирования (7)-(13) имеем задачу динамического программирования:

$$W_k = \min_{x_k, n_k} \{W_{k-1} + (x_k (1 + r(n_k))^{n_k} - x_k)\} \quad (14)$$

$$s_k = s_{k-1} - d_k + p_k + x_k - \sum_{i=1}^{k-1} Y_i \quad (15)$$

$$x^{\min} \leq x_k \leq x^{\max}, k = \overline{1, N} \quad (16)$$

$$n^{\min} \leq n_k \leq n^{\max}, k = \overline{1, N} \quad (17)$$

$$t_k \in \{1, 2, 3, 4\}, k = \overline{1, N} \quad (18)$$

$$s_0 = Q \quad (19)$$

$$s_k \geq 0, k = \overline{1, N} \quad (20)$$

Решением задачи (14)-(20) является определение таких значений  $x_k$ ,  $n_k$  и  $t_k$ , которые удовлетворяют условиям (15)-(20) и доставляют минимум (14). Задача математического программирования (14)-(20) относится к классу задач с аддитивной целевой функцией, для решения которых наиболее часто применяется алгоритм «киевский веник». Основное содержание алгоритма состоит в формулировке правил последовательного сужения множества конкурентоспособных вариантов. Однако, задача (14)-(20) не удовлетворяет принципу оптимальности Беллмана, который гласит, что характер траектории, переводящей систему из положения  $(x_i, t_i)$  в положение  $(x_k, t_k)$  ( $i < k$ ), не должен зависеть от состояния системы в моменты, предшествующие  $t_i$ . Это связано с тем, что управления, осуществляемые на  $i$ -м этапе, влияют на систему ограничений (14)-(20) посредством изменения переменной  $Y_k$  ( $k > i$ ).

Таким образом, наиболее рациональным методом решения данной задачи динамического программирования является метод последовательного анализа

вариантов. Совокупность величин  $x_k$ ,  $n_k$  и  $t_k$  образуют некоторый вариант кредитования проекта на  $k$ -м шаге. В рассматриваемом случае все указанные переменные являются дискретными, поэтому множество вариантов для каждого этапа конечно. Пусть для  $k$ -го этапа таких вариантов  $M_k$ . Величина  $M_k$  не является постоянной и существенно зависит от развития ситуации в предыдущих этапах. Максимальное значение  $M_k$  определяется пределами изменения размера кредита, пределами изменения сроков кредитования, шагом квантования размера кредита и набором фиксированных параметров. В ходе решения мощность множества вариантов уменьшается, поскольку накладываются ограничения (16)-(20). Так, например, часть вариантов отбрасывается за счет ограничений (17) и (20), которые определяют нижнюю границу величины кредита в  $k$ -м этапе.

**Функциональные возможности СППР.** В работах [4-6] представлена концепция СППР, а также ряд моделей построения схем финансирования с привлечением инвестиционных средств от различных источников. В данной работе предлагается программная реализация данной системы. На первом этапе разработки программного комплекса были сформированы функциональные требования к системе. Все функции можно разбить на два класса:

1. Функции, направленные на расчет оптимальной схемы финансирования инвестиционного проекта.

2. Функции, направленные на обеспечение удобства взаимодействия пользователя с программой.

Общая функциональность программы отражена на диаграмме вариантов использования на рис. 1.

В программе поддерживаются следующие функции, ненаправленные прямо на решение поставленной задачи:

- редактирование (сохранение, удаление) данных о предприятиях;
- редактирование (сохранение, удаление) данных об инвесторах;
- редактирование (сохранение, удаление) данных о банках и вариантах кредитования инвестиционных проектов, предоставляемые этими данными;
- редактирование (сохранение, удаление) данных о проектах;
- редактирование (сохранение, удаление) параметров модели расчета;
- сохранение отчета о проведенных расчетах в формате HTML.

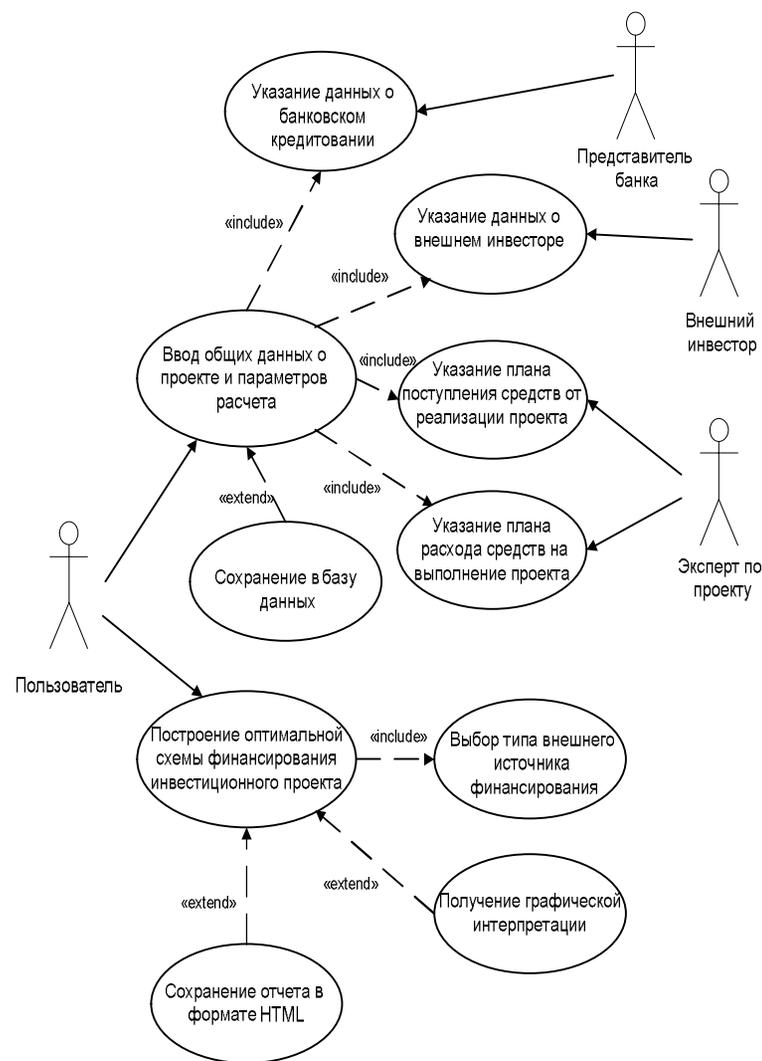


Рис. 1. Диаграмма вариантов использования СППР

**Структура программного комплекса.** Данный программный комплекс может быть установлен на компьютере под управлением любой операционной системы, поддерживаемой технологией Java. Для запуска программы необходимо, чтобы была установлена виртуальная машина Java

версии 1.5 или старше. Программное обеспечение представлено в комплекте из шести файлов, описание которых приведено в табл. 1.

Таблица 2

Таблица 1

Комплектность программного обеспечения

№п/п	Имя файла	Описание
1	pio.jar	Java-архив, содержащий все классы программы. Содержит главный класс org.khpi.asu.ui.PioMain
2	lib/jcommon-1.0.16.jar	Библиотека вспомогательных классов.
3	lib/jfreechart-1.0.13.jar	Библиотека классов для построения графиков в Swing/AWT программах
4	lib/sqlitejdbc-v056.jar	Java-архив, содержащий JDBC драйвер для работы с СУБД SQLite
5	db.properties	Конфигурационный файл
6	piodb.db	Файл базы данных

Запуск приложения осуществляется путем выполнения файла pio.jar командой javaw -jar pio.jar. После запуска приложения на экране открывается главное окно программы. Оно содержит главное меню программы, позволяющее получить доступ к другим окнам и функциям программы, а также список сохраненных в базе данных параметров моделей расчета схем финансирования проектов. С помощью разработанного интерфейса можно редактировать данные, занесенные в базу данных. В результате реализации предложенных моделей схем финансирования на экран выводятся итоговые и промежуточные расчеты.

Каждой экранной форме в структуре программного обеспечения соответствует файл, содержащий класс, описывающий данную форму. Кроме этого, файлы содержат функции направленные на обеспечение удобства взаимодействия пользователя с программой, и функции обработки событий форм. К таким событиям относятся: события, вызываемые действиями пользователя (нажатие кнопок, ввод/изменение данных, изменение размеров рабочей области окна и т.п.) и события, вызываемые командами из других форм (изменение содержания или оформления формы, открытие/закрытие окна и др.).

Перечень классов, которые реализуют аналитическую часть, приведен в табл. 2. В программе реализованы классы для взаимодействия с базой данных, записи ошибок возникающих в программе для последующего их анализа, и классов, реализующих экранные формы для взаимодействий с пользователем. Краткое описание классов, не относящихся непосредственно к реализации алгоритма расчета оптимальной схемы финансирования предприятия, представлено в табл. 3.

Описание классов, реализующих расчетную функциональность СПИР

№ п/п	Название класса	Описание
1	org.khpi.asu.math.Simplex	Класс, реализующий решение ЗЛП методом последовательного улучшения плана
2	org.khpi.asu.math.fuzzy.FuzzyLinearOptimization	Класс, реализующий решение ЗЛП с нечеткими коэффициентами
3	org.khpi.asu.math.fuzzy.FuzzyNumber	Класс, описывающий нечеткое число треугольного вида
4	org.khpi.asu.investments.InvestmentsOptimization	Класс, реализующий расчет оптимальной схемы финансирования предприятия с привлечением прямых инвестиций
5	org.khpi.asu.crediting.CreditsOptimization	Класс, реализующий расчет оптимальной схемы финансирования предприятия с привлечением кредита
6	org.khpi.asu.investments.criteria.InvestorNpvCriterion	Класс, описывающий критерий максимизации прибыли инвестора
7	org.khpi.asu.investments.criteria.EnterpriseNpvCriterion	Класс, описывающий критерий максимизации прибыли предприятия
8	org.khpi.asu.investments.criteria.CapitalCostsCriterion	Класс, описывающий критерий минимизации суммарной стоимости затраченного капитала
9	org.khpi.asu.data.entities.Project	Класс, описывающий проект и его параметры
10	org.khpi.asu.data.entities.Investor	Класс, описывающий инвестора
11	org.khpi.asu.data.entities.Bank	Класс, описывающий банк и варианты кредитования, предоставляемые этим банком
12	org.khpi.asu.crediting.Credit	Класс, описывающий кредит, выданный данным банком данному предприятию для финансирования указанного проекта
13	org.khpi.asu.data.entities.Enterprise	Класс, описывающий предприятие
14	org.khpi.asu.data.entities.InvestingModel	Класс, описывающий параметры математической модели для расчета оптимальной схемы финансирования предприятия с привлечением прямых инвестиций
15	org.khpi.asu.data.entities.CreditingModel	Класс, описывающий параметры математической модели для расчета оптимальной схемы финансирования предприятия с привлечением банковских кредитов
16	org.khpi.asu.data.entities.PaymentScheme	Перечисление, содержащее описание схем погашения кредитов

Таблица 3

Описание классов, реализующих экранные формы для взаимодействий с пользователем

№ п/п	Название класса	Описание
1	org.khpi.asu.ui.PioMain	Главное окно программы. Содержит список сохраненных математических моделей
2	org.khpi.asu.ui.EditInvOrEntFrame	Окно редактирования данных о предприятиях и инвесторах
3	org.khpi.asu.ui.EditProjectsFrame	Окно редактирования данных о проектах
4	org.khpi.asu.ui.EditBankFrame	Окно редактирования данных о банках и предоставляемых ими схемах кредитования инвестиционных проектов
5	org.khpi.asu.ui.ModelTypeSelector	Окно выбора типа источника внешнего финансирования перед началом построения оптимальной схемы финансирования инвестиционного проекта
6	org.khpi.asu.ui.InvestringModelFrame	Окно редактирования параметров математической модели для построения оптимальной схемы финансирования инвестиционного проекта с привлечением прямых инвестиций
7	org.khpi.asu.ui.CreditingModelFrame	Окно редактирования параметров математической модели для построения оптимальной схемы финансирования инвестиционного проекта с привлечением банковского кредита
8	org.khpi.asu.ui.ModelsTableModel	Модель таблицы, содержащей список математических моделей
9	org.khpi.asu.ui.InvestingOptimizationResultsFrame	Окно для отображения результатов расчета оптимальной схемы финансирования инвестиционного проекта с привлечением прямых инвестиций. Содержит как табличное, так и графическое представление данных
10	org.khpi.asu.ui.CreditingOptimizationResultsFrame	Окно для отображения результатов расчета оптимальной схемы финансирования инвестиционного проекта с привлечением банковского кредита. Содержит табличное и графическое представление данных
11	org.khpi.asu.ui.AboutBox	Окно, содержащее информацию о программе
12	org.khpi.asu.data.DBContext	Класс, реализующий взаимодействие с базой данных
13	org.khpi.asu.data.DBManager	Класс, реализующий загрузку, сохранение и удаление данных из базы данных
14	org.khpi.asu.logging.Logger	Класс, реализующий запись ошибок программы и отладочной информации в специальный файл

Аппаратное обеспечение, необходимое для использования программного продукта, определяется из следующих соображений, с одной стороны, оно предназначено для широкого использования и чрезмерные системные требования могут значительно сократить возможности его переносимости; с другой стороны, аппаратное обеспечение должно обеспечивать комфортную работу с ним. Поэтому наиболее подходящим является компьютер 6-го поколения или более современный. Минимально необходимая тактовая частота процессора составляет минимум 533 МГц, а рекомендуемая 733 МГц и выше. В ходе тестирования было установлено, что требуется около 40 Мб свободной оперативной памяти. Общий объем требуемой оперативной памяти при этом существенно зависит от используемой операционной системы. Непосредственно для работы программного обеспечения необходимо около 10 Мб свободного места на жестком диске.

Нужно отметить, что в каждом конкретном случае объем диска должен определяться отдельно, поскольку это существенно зависит от используемой операционной системы и другого программного обеспечения, установленного на компьютере.

**Выводы.** Таким образом, спроектированный программный комплекс реализует систему поддержки принятия финансовых решений при управлении инвестиционными проектами. В частности, реализует процесс построения схем финансирования проектов с привлечением инвестиционных средств из различных источников. Разработанный программный продукт может использоваться как автономное приложение, так и может быть интегрирован в систему управления проектами или в систему управления предприятия. Предлагаемая СППР будет апробирована для построения схем финансирования инвестиционного проекта по строительству мусороперерабатывающего завода в г. Харькове, а также ряда других проектов.

**Список литературы:** 1. Benjamin C. Engineering for Business: Theory and Cases. - University Press of America, 2007.- 278p. 2. Вязовой В. С. Сравнительная таблица Open Plan Professional, Primavera, Microsoft Project // [www.project.km.ru](http://www.project.km.ru), 15.08.2009. 3. Мировой опыт внедрения//<http://www.projectmanagement.ru/pvsl.asp>, 13.11.2009. 4. Годлевский М. Д., Москаленко В. В., Кондращенко В. В. Математическая модель расчета оптимальной схемы финансирования инвестиционного проекта с привлечением внешних инвестиций.// Вестник НТУ «ХПИ» «Системный анализ, управление и информационные технологии».- Харьков: НТУ «ХПИ», 2006. – №19.- С. 151-157. 5. Годлевский М. Д., Москаленко В. В., Кондращенко В. В. Динамическая модель процесса финансирования инвестиционного проекта // Системный анализ та інформаційні технології: Матеріали ІХ Міжнародної науково-технічної конференції (15-19 травня 2007р., Київ). – К.:НТУУ «КПІ», 2007. –С. 99. 6. Годлевский М. Д., Москаленко В. В., Кондращенко В. В. Система поддержки принятия решений процесса финансирования инвестиционного проекта // Вісник Національного технічного університету «Харківський політехнічний інститут»: Тематичний випуск «Системний аналіз, управління та інформаційні технології». – Харків: НТУ «ХПІ». – 2007. – №5. – С. 75-89.

Поступила в редколлегию 07.12.09

**В. В. КРЮЧКОВСКИЙ**, канд. физ.-мат. наук, декан ф-та  
международного менеджмента Херсонского нац. техн. ун-та (г. Херсон),  
**Н. А. БРЫНЗА**, аспирантка ХНУРЭ, каф. СТ,  
**А. Х. БАДДУР**, аспирант ХНУРЭ, каф. СТ

### АНАЛИЗ АДЕКВАТНОСТИ ВЗАИМНОЙ ТРАНСФОРМАЦИИ НЕОПРЕДЕЛЕННОСТЕЙ ПРИ ВЫЧИСЛЕНИИ СКАЛЯРНЫХ ИНТЕРВАЛЬНЫХ ЗНАЧЕНИЙ ПОЛЕЗНОСТИ АЛЬТЕРНАТИВ

У статті розглянуто можливість приведення різномірних невизначеностей (статистичних, нечітких, інтервальних) до єдиного базисного виду при обчисленні скалярних багатofакторних оцінок корисності альтернативних рішень в умовах невизначеності. На основі тестового моделювання показано, що вибір базисної форми невизначеності не впливає на відношення порядку альтернатив і незначно змінює силу уподобання, але істотно впливає на величину інтервалу невизначеності.

В статье рассмотрена возможность приведения разнородных неопределенностей (статистических, нечетких, интервальных) к единому базисному виду при вычислении скалярных многофакторных оценок полезности альтернативных решений в условиях неопределенности. На основе тестового моделирования показано, что выбор базисной формы неопределенности не влияет на отношение порядка альтернатив и незначительно изменяет силу их предпочтения, но существенно влияет на величину интервала неопределенности.

The article discusses the possibility of bringing disparate uncertainties (statistical, fuzzy, interval) to a single base type in the calculation of the scalar multivariate evaluations of the usefulness of alternative solutions in the face of uncertainty. Based on the test simulation showed that the choice of basic forms of uncertainty does not affect the order of the alternatives, and slightly modifies the strength of their preferences, but significantly affect the value of the interval of uncertainty.

**Введение.** Обязательным этапом любой целенаправленной деятельности является принятия решений. При этом не только неверные, но и не эффективные решения приводят к потерям или нерациональному использованию финансовых, материальных, социальных, экологических и т.д. ресурсов. В связи с этим проблема разработки научно – обоснованной методологии принятия эффективных решений является одной из актуальных проблем современности.

По определению В. М. Глушкова [1] необходимыми условиями эффективности любого решения являются: своевременность, полнота (комплексность), оптимальность. При этом под полнотой понимается как можно более полный и глубокий учет всех факторов, определяющих не только текущие, но и отдаленные последствия принимаемых решений. Такой подход по необходимости предопределяет многокритериальность задачи принятия решений.

Другим следствием стремления к полноте является возрастание размерности и сложности задачи. Это связано с необходимостью учета

большого количество не только прямо, но и косвенно влияющих на решение факторов и идентификации их взаимосвязи. В совокупности, это приводит к резкому возрастанию неопределенности исходной информации.

Подводя итог, можно сделать вывод, что обязательным условием методологии принятия эффективных решений является необходимость, в общем случае, учета многокритериальности постановки задачи и неопределенности исходной информации.

**Обзор состояния проблемы.** Процесс принятия решения в общем случае можно структурировать на следующие этапы: формирование цели, определение множества допустимых решений  $X$ , задание метрики, в которой оценивается качество (эффективность) допустимых решений, определение экстремального (оптимального) по качеству решения.

Задание метрики оценки качества допустимых решений связано с формированием некоторого критерия, позволяющего ранжировать альтернативы. Как правило, не удается выбрать единственный критерий, который достаточно полно и однозначно характеризовал бы «качество» решений. В общем случае, приходится формировать некоторое множество частных, разнородных по смыслу, размерности, направлению доминирования, то есть противоречивых критериев, каждый из которых характеризует отдельное или комплекс качеств, а в совокупности они достаточно полно и однозначно характеризуют решение в целом.

Обозначим кортеж таких критериев, как

$$K(x) = \langle k_i(x) \rangle, \quad i = \overline{1, n}, \quad (1)$$

где  $k_i(x)$  -  $i$ -й частный критерий,  $n$  – число частных критериев.

Тогда в общем случае задача определения оптимального решения

$$x^o = \arg \operatorname{extr}_{x \in X} \langle k_i(x) \rangle, \quad i = \overline{1, n} \quad (2)$$

является некорректной по Адамару [2], так как определяет не единственное решение, а некоторое подмножество недоминируемых решений  $X^c \in X$ , которые принято называть множеством компромиссных или Парето - оптимальных решений [3].

Выбор из подмножества  $X^c$  единственного решения связан с необходимостью регуляризации задачи (2), которая заключается в формировании некоторого дополнительного эвристического правила (модели), известного как принцип (схема) выбора компромиссного решения  $x^* \in X^c$ . Общей чертой этих схем является сведение исходной задачи многокритериальной оптимизации к задаче однокритериальной скалярной оптимизации или последовательности таких задач. Таких методов много, достаточно назвать принцип главного критерия [4], методы последовательной

оптимизации, функционально-стоимостного анализа, анализа иерархий и т.д. Несмотря на их разнообразие, все они по сути дела являются частными случаями подхода, основанного на теории полезности [5]. Теория полезности базируется на гипотезе [6], о том, что для любого решения, которое характеризуется кортежем разнородных частных критериев  $k_i(x)$ ,  $i = \overline{1, n}$ , существует обобщенная скалярная многофакторная оценка эффективности (полезности)

$$P(x) = F[(A, k_i(x_j)], \quad i = \overline{1, n}, \quad (3)$$

где  $F$  - оператор, определяющий структуру, а кортеж  $A$  – параметры модели оценивания.

Процедура многофакторного оценивания является субъективной интеллектуальной процедурой, поэтому носителями исходной информации, необходимой для структурно-параметрической идентификации ее модели является специалисты (эксперты) в различных проблемных областях, а основным методом получения первичной информации – метод экспертного оценивания. Субъективизм метода экспертного оценивания и широта проблемно – ориентированных задач привели к тому, что в настоящее время на практике используются несколько альтернативных моделей многофакторного оценивания (3). Наиболее широко известны аддитивная [7]

$$P(x_j) = \sum_{i=1}^n a_i k_i^n(x_j), \quad (4)$$

где  $k_i^n(x)$  - нормализованные, т.е. приведенные к безразмерному виду, единому интервалу [0, 1] возможных значений и одинаковому направлению доминирования частные критерии;  $a_i$  - безразмерные коэффициенты относительной важности нормализованных частных критериев.

По определению, для коэффициентов  $a_i$  должны выполняться следующие требования

$$0 \leq a_i \leq 1, \quad \forall i = \overline{1, n}, \quad \sum_{i=1}^n a_i = 1. \quad (5)$$

Кроме того, используются мультипликативная, аддитивно-мультипликативная модели и модель Кобба – Дугласа. Однако анализ показывает, что все перечисленные модели оценивания являются частными случаями (фрагментами) полинома Колмогорова – Габора [8]:

$$P(x) = a_0 + \sum_{i=1}^n a_i k_i(x) + \sum_{i=1}^n \sum_{j=1}^n a_{ij} k_i(x) k_j(x) + \sum_{i=1}^n \sum_{j=1}^n \sum_{l=1}^n a_{ijl} k_i(x) k_j(x) k_l(x). \quad (6)$$

В дальнейшем будем полагать, что модель оценивания (3) представляет собой некоторый фрагмент полинома Колмогорова – Габора, который идентифицирован непосредственно методом экспертного оценивания или методом компараторной идентификации [9]. Таким образом, будем полагать, что модель многофакторного оценивания, т.е. вычисления скалярной полезности альтернатив, задана.

Независимо от метода структурно – параметрической идентификации модель скалярного оценивания полезности решений (3) содержит неопределенности. Эти неопределенности порождаются НЕ-факторами [10], т.е. неполнотой знаний, неточностью измерения частных критериев, неточностью идентификации весовых коэффициентов и т.д.

Примем, что любой параметр или переменная модели многофакторного скалярного оценивания (функции полезности) заданы в интервальном виде [11]. Это означает, что исходные данные представляются интервальными числами  $[D_l, D_r]$ , задаваемые левыми  $D_l$  и правыми границами  $D_r$  на числовой оси. Размах интервала количественно характеризует степень неопределенности величины. В частном случае, при  $D_l = D_r$ , величина задается точечным детерминированным значением.

Важнейшей качественной характеристикой неопределенности является информация о характере распределения возможных значений внутри интервала. По этому признаку можно выделить:

- статистическую (вероятностную) неопределенность;
- нечеткую (представленную в виде нечеткого множества) неопределенность;
- интервальные величины.

В случае статистической неопределенности характер распределения значений на интервале задается функцией плотности распределения вероятности и соответствующими статистическими параметрами: математическим ожиданием, дисперсией и т.д. Это наиболее объективная информация о характере распределения значений, так как основой ее получения являются результаты обработки многократных экспериментальных наблюдений [12].

Во многих случаях статистическая оценка не может быть получена по причинам отсутствия представительной выборки наблюдений, ее статистической неоднородности, или когда анализируемая величина принципиально не может быть интерпретирована как случайная. Примером могут служить нечеткие числа «около 5», «приблизительно 2», «меньше 3» и многие другие нечеткие лингвистические утверждения. В этом случае характер распределения возможных значений на интервале может быть описан функцией принадлежности нечеткому множеству [13]. Эта информация полностью субъективна, так как отражает знания и опыт одного или группы экспертов.

В третьем случае, у аналитика отсутствует как объективная, так и субъективная информация о характере распределения возможных значений на интервале. Такие неопределенности называются интервальными величинами [14].

Подводя итог, отметим, что с учетом введенных определений, модель скалярного многофакторного оценивания полезности альтернативных решений (3) будет иметь вид

$$\bar{P}(x) = F[(\bar{A}, \bar{k}_i(x_j)], \quad i = \bar{1}, n, \quad (7)$$

где знаком « $\bar{\phantom{x}}$ » отмечены интервальные неопределенные величины различного вида.

Особенность модели (7) заключается в том, что результат оценивания является интервальным числом. Вместе с этим, конечная цель процедуры принятия решений заключается в выборе конкретного точечного решения.

Общепринятой является методология принятия решений в условиях неопределенности, которая предусматривает вычисления интервальных значений полезности решений по модели (7) и последующий выбор точечного решения как компромисса между оптимистическим и пессимистическим решениями, например, на основе VaR технологий [15].

Обязательным этапом реализации методологии принятия решений в условиях неопределенности является вычисление интервальных значений многофакторной скалярной оценки полезности альтернативных решений  $x \in X$ . Эта задача не вызывает принципиальных затруднений в том случае, если все неопределенности относятся к одному виду по информации о характере распределения значений на интервале. Для каждого вида неопределенности (статистической, нечеткой, интервальных величин) определены специализированные правила выполнения арифметических операций сложения и умножения, которые необходимы для вычисления полезности  $P(x)$ . В том случае, если в модель входят разнородные неопределенности (это наиболее часто встречающиеся ситуации), возникает задача их взаимной трансформации с целью приведения к одному базисному виду.

**Цель и методика исследования.** Целью является определение методом тестового моделирования принципиальной возможности и степени корректности взаимной трансформации различных видов интервальных неопределенностей к однородному виду – статистическому, нечеткому или к виду интервальных величин. При этом, под корректностью понимается сохранение отношения порядка по полезности на множестве альтернативных решений, соотношение величины интервальных оценок полезности альтернатив, полученных при различных исходных формах задания неопределенностей, силы предпочтения (расстояния между альтернативными решениями по полезности).

Методика тестирования заключается в следующем. Для того, чтобы получить некоторый независимый базис (внешнее дополнение)[8], относительно которого производится сравнение, формируется некоторая эталонная ситуация. В качестве такого эталона принята детерминированная ситуация, когда все параметры и переменные модели вычисления скалярной многофакторной оценки полезности альтернатив (7) представлены детерминированными точечными значениями. По модели (7) для этих исходных данных вычисляются значения полезности  $P(x)$ , по ним устанавливается отношение порядка

$$x_1 \mathbf{f} x_2 \mathbf{f} \dots \mathbf{f} x_n, \quad (8)$$

и сила предпочтительности, т.е. расстояние между смежными альтернативами по величине оценок их полезности

$$\Delta P_{12} = P(x_2) - P(x_1) \quad (9)$$

Затем исходные точечные детерминированные значения параметров и переменных модели трансформируются в интервальные неопределенности. При этом, исходные точечные значения принимаются, соответственно, в качестве математического ожидания для статистической неопределенности, модального значения для нечетких множеств и центра интервальной величины. Величина интервала во всех случаях принимается одинаковой.

При вероятностной неопределенности вычисления производятся на основании статистических параметров – математического ожидания и дисперсии. Для перехода от интервальных значений к статистическим параметрам использовались следующие соотношения.

Оценка математического ожидания на основе данных о границах интервала для нормального закона распределения определяется следующим образом[12]:

$$M = \frac{(a+b)}{2}; \quad (10)$$

среднеквадратическое отклонение

$$s = \frac{b-a}{6}; \quad (11)$$

дисперсия вычисляется как

$$D = s^2. \quad (12)$$

Соответственно для равновероятного закона распределения:

$$M = \frac{(a+b)}{2}; \quad (13)$$

$$D = \frac{(b-a)^2}{12}. \quad (14)$$

Арифметические операции со статистическими параметрами выполняются по правилам [13]:

$$M(X_1 + X_2) = M(X_1) + M(X_2); \quad (15)$$

$$D(X_1 + X_2) = D(X_1) + D(X_2); \quad (16)$$

$$M[XY] = M[X]M[Y]; \quad (17)$$

$$D[XY] = D[X]D[Y]; \quad (18)$$

$$cM[X] = M[cX]; \quad (19)$$

$$cD[X] = D[c^2X]; \quad (20)$$

Для случая, когда частные критерии и весовые коэффициенты представлены в виде нечетких множеств (нечетких чисел в R,L – форме [16]), вычисление функции полезности производилась по следующим формулам сложения и умножения [16, 17]

$$(a_1, b_1, b_1)_{LR} + (a_2, b_2, b_2)_{LR} \sim (a_1 + a_2, b_1 + b_2, b_1 + b_2)_{LR} \quad (21)$$

$\forall A$ , таких, что,  $m_A, m_B \in F(R^+)$ ,  $a_1 > 0$ ,  $a_2 > 0$ ,

$$(a_1, b_1, b_1)_{LR} * (a_2, b_2, b_2)_{LR} \sim (a_1 a_2, a_1 b_2 + a_2 b_1, a_1 b_2 + a_2 b_1)_{LR}, \quad (22)$$

где  $a_1, a_2$  – левые границы нечетких множеств,  $b_1, b_2$  – правые границы нечетких множеств,  $b_1, b_2$  – модальные значения, при которых функция принадлежности функции равна 1.

Аналитические правила выполнения арифметических операций с интервальными величинами имеют вид [14]:

$$\begin{cases} A + B = [a_1 + a_2, b_1 + b_2], \\ A \cdot B = [\min\{a_1 a_2\}, \{a_1 b_2\}, \{a_2 b_1\}, \{b_1 b_2\}], \\ \max\{a_1 a_2\}, \{a_1 b_2\}, \{a_2 b_1\}, \{b_1 b_2\}. \end{cases} \quad (23)$$

Для того, чтобы получить представительные результаты, позволяющие сделать корректные выводы, тестирование производилось для задач

оценивания различной размерности по числу частных критериев (n=2, 4, 7) и для различных моделей (линейных и нелинейных) функции полезности. При этом для каждого случая рассматривались следующие виды неопределенности

- статистическая (нормальный и равновероятностный закон распределения);
- нечеткая (треугольная форма функции принадлежности);
- интервальные величины.

Множество альтернативных решений во всех случаях равно 7.

Для каждого тестового примера вычислялись интервальные значения функции полезности альтернативных решений, отношение порядка на множестве альтернатив и соответствующая сила предпочтения. При этом отношение порядка определялось по значениям полезности соответствующей математическому ожиданию, моде и центральному значению соответственно.

**Заключение.** По методике, описанной выше, были протестированы линейные и нелинейные функции полезности вида:

$$P(x) = a_1 k_1(x) + a_2 k_2(x) + a_3 k_1(x) k_2(x),$$

$$P(x) = a_1 k_1(x) + a_2 k_2(x) + a_3 k_3^2(x) + a_4 k_2(x) k_4(x).$$

В силу громоздкости, результаты численного моделирования в статье не приведены, но необходимо отметить, что отношения порядка альтернатив во всех случаях соответствуют эталонным значениям. Анализ результатов моделирования позволяет сделать следующие выводы.

При приведении к единому базису разнородных видов неопределенности в принципе, в качестве базисной может быть использована любая форма. Это обусловлено тем, что выбор формы представления не влияет на отношения порядка альтернатив: лучшая альтернатива во всех случаях остается экстремальной, а сила предпочтения изменяется не значительно (в среднем на 10%).

Очень важной характеристикой является величина интервала неопределенности полезности альтернатив. Во всех случаях минимальный интервал соответствует статистической форме представления исходной информации. При нормальном законе распределения возможных значений интервал на 70% меньше, чем при равновероятностном законе и еще меньше для других видов неопределенности. Это легко объяснить, так как исходная информация в этом случае получена на реальной экспериментальной выборке, в то время как все другие формы отражают субъективные представления.

На третьем месте по точности находится оценки полезности, полученные при задании частных критериев в виде интервальных величин.

Неожиданным является тот факт, что самая большая интервальная неопределенность  $P(x)$  соответствует нечеткой форме представления исходной информации. Это можно объяснить тем, что нечеткие множества не предназначены для формализации нечетких чисел, а ориентированы на формализацию качественной информации представленной в виде нечетких лингвистических высказываний.

Такие результаты справедливы для всех полиномиальных форм (как линейных, так и нелинейных) модели функции полезности.

На основе сказанного можно сделать вывод, что наиболее предпочтительной базовой формой является статистическая неопределенность с равномерным законом распределения. Это обусловлено тем, что приведение к наиболее эффективному нормальному закону распределения требует информации, которая реально отсутствует. Кроме того, нечеткие величины принципиально не могут быть интерпретированы как случайные. В то же время, равновероятностный закон распределения можно интерпретировать не как случайную величину, как ситуацию, в которой отсутствует информация о предпочтениях конкретных значений, т.е. они все равновозможны (равновероятны).

**Список литературы:** 1. Глушков В.М. Введение в АСУ/ В. М. Глушков. – Киев: Техника, 1972. – 312 с. 2. *Математический энциклопедический словарь*/ под ред. Ю.В. Прохорова. – М.: Сов. энциклопедия, 1988. – 250с. 3. *Подinovский В.В.* Парето-оптимальные решения многокритериальных задач/ В.В. Подinovский, В.Д. Ногин. – М.: Наука, 1982. – 254с. 4. *Петров Е.Г.* Методы і засоби прийняття рішень у соціально-економічних системах/ Е.Г. Петров, М.В. Новожилова, І.В. Гребенник. – Київ: Техніка, 2004. – 256с 5. *Фишберн П.* Теория полезности для принятия решений/ П. Фишберн. – М.: Наука, 1978. – 352с. 6. *Нейман Дж.* Теория игр и экономическое поведение/ Дж. Нейман, О. Моргенштерн. – М.: Наука, 1970. – 124с. 7. *Штойер Р.* Многокритериальная оптимизация. Теория, расчет и приложения/ Р. Штойер. – М.: Радио и связь, 1992. 504с. 8. *Ивахненко А.Г.* Самоорганизация прогнозирующих моделей/ А.Г. Ивахненко, И.А. Мюллер. – К.: Техника, 1985. – 233с. 9. *Петров К.Э.* Компараторная структурно-параметрическая идентификация моделей скалярного многофакторного оценивания/ К.Э. Петров, В.В. Крючковский. – Херсон: Олди-плюс, 2009. – 294с. 10. *Нариньяни А.С.* НЕ-факторы: неоднозначность (доформальное исследование)/ А.С. Нариньяни // *Новости искусственного интеллекта.* – 2003.–№5. с.58-69. 11. *Стернин М.Ю.* Метод представления знаний в интеллектуальных системах поддержки экспертных решений/ М.Ю. Стернин, Г.И. Шепелев// *Новости искусственного интеллекта.* 2003. – №4(58), с.58-69. 12. *Вентцель Е.С.* Теория вероятности и ее инженерное приложение/ Е.С. Вентцель, Л.А. Овчаров – М.: Высшая школа, 2000.–480с. 13. *Заде Л. А.* Понятие лингвистической переменной и его применение к принятию приближенных решений/ Л. Заде – М.: «Мир», 1976. 14. *Алефельд Г.* Введение в интервальные вычисления: Пер. с англ./ Алефельд Г., Херцбергер Ю –М. Мир, 1987. – 360 с. 15. *Бідюк П.І.* Методи прогнозування/ П.І. Бідюк, О.С.Меняйленко, О.В. Половцев. – Луганськ: «Алма-матер». 2008. – Том1. – 302с. 16. *Раскин Л.Г.* Нечеткая математика. Основы теории. Приложения / Раскин Л.Г., Серая О.В. - Харьков: Парус, 2008. - 352 с. 17. *Борисов А.Н.* Обработка нечеткой информации в системах принятия решений/ А.Н. Борисов, А.В. Алексеев, Г.В. Меркурьев и др. – М., Радио и связь, 1989.

Поступила в редакцию 18.12.09

УДК 519.1

**К. О. СОЛОВЬОВА**, д-р техн. наук, зав. каф СІ ХНУРЕ,  
**В. В. МОВЧАН**, стажист-дослідник ХНУРЕ

## РОЗРОБКА МОДЕЛІ ПРОГРАМНОГО ЗАСОБУ ВІЗУАЛІЗАЦІЇ МЕРЕЖЕВИХ СТРУКТУР

Розглядаються сучасні методи та існуючі програмні засоби візуалізації мережних структур. Визначаються вимоги до програмного засобу з візуалізації мережних структур. Визначаються недоліки існуючих програмних засобів. Розроблюється модель програмного засобу, що відповідає означеним вимогам. Створюється програмний засіб на основі розробленої моделі.

Рассматриваются современные методы и существующие программные средства визуализации сетевых структур. Определяются требования к программному средству визуализации сетевых структур. Определяются недостатки существующих программных средств. Разрабатывается модель программного средства, отвечающего определенным требованиям. Создается программное средство на основе разработанной модели.

The article deals with modern methods and existing software visualization of network structures. Identify requirements for the visualization software network structures. Identify the shortcomings of existing software. The author develops a model of software tools that meet certain requirements. It creates a software tool based on the developed model.

**Постановка проблеми.** Ускладнення усіх видів економічної діяльності в сучасних умовах призвело до зміщення акцентів розвитку спільноти з матеріальних на інформаційні. Разом із цим, значно зросла роль аналізу інформації та гостро постала проблема забезпечення аналітиків зручними та доступними засобами обробки та аналізу інформації.

Природним засобом представлення знань є представлення їх у вигляді графу [2]. Теорія графів набула широкого використання в таких несхожих сферах, як економіка, системологія [7], теорія управління, маркетинг, інформатика, хімія, програмування, логістика, схемотехніка та безліч інших [2, 3]. Практично, за допомогою теорії графів може бути, в тій чи мірі, представлена будь-яка інформація чи знання.

Проблему забезпечення аналітика, інструментом, що базується на теорії графів намагаються вирішити не перший рік - перші програмні реалізації такого інструменту з'явилися на початку дев'яностих років [1]. З того часу була створена чисельна множина різноманітних програм, які в тій чи іншій мірі виконують своє призначення бути засобом для відображення графових структур. Кожна з цих реалізацій має свої недоліки та переваги. Але ці програми мають один великий спільний недолік, який полягає в тому, що вони не є доступними, тобто мають високу ціну та не є крос-платформними. Крім того, у безкоштовних реалізаціях програмних засобів часто не передбачена підтримка кирилических шрифтів та використовуються спеціальні недокументовані формати вихідних даних, що ускладнює організацію

передачі даних у інші програми. Беззаперечно, існують програми, які можливо завантажити з мережі Internet. Та ці програми, зазвичай, мають або дуже обмежений функціонал, або не є безкоштовними і можуть бути використовувані без оплати лише протягом ознайомчого періоду.

Таким чином, актуальною є задача розробки моделі програмного засобу та її програмна реалізація, що не має означених недоліків та може використовуватися в Україні для вирішення проблеми забезпечення аналітика зручним засобом з візуалізації інформації.

**Мета статті.** Розробка моделі програмного засобу та її програмна реалізація, що не матиме зазначених вище недоліків, а саме: буде крос-платформною, підтримуватиме кириличні шрифти, буде використовувати загальновідомий формат вихідних даних XML. Крім того, програмний засіб має бути відкритий.

**Аналіз останніх досліджень і програмних засобів.** Наукоємні продукти з використанням методів візуалізації інформації існують на світовому ринку останні 15 років. Є фірми, що надають бібліотеки і програмні комплекси, орієнтовані на візуалізацію графів загального призначення (Tom Sawyer software, ILOG software, Algorithmic Solutions Software GmbH, yWorks), а також системи програмного забезпечення (Imagix Corporation, Absint та ін.) [1]. «В додаток з'являється все більше фірм, що спеціалізуються на візуалізації бізнес-інформації, необхідної аналітикам різних підприємств та орієнтованої на спеціалізацію тих чи інших підприємств (Enterprise Solutions). Однією зі старіших компаній цього напрямку є фірма Inxight Software, Inc, що надає засоби візуалізації інформації для фінансових і біологічних фірм. З'явилися компанії, що надають на ринок продукти, які використовують нові методи в галузі візуалізації інформації, так звані методи візуальної аналітики. В зростаючій родині інструментів візуальної аналітики квітне програмне забезпечення фірми HiveGroup, яка використовує візуалізацію ієрархічних даних на ґрунті карти дерева (Treemap) і надає організаціям потрібний їм щоденний моніторинг складної діяльності за участю тисяч продуктів, проєктів та продавців» [1].

**Визначення вимог до програмного засобу.** Проблема розміщення графу є невід'ємною частиною теорії графів та проявляється при роботі навіть із невеликими графами, та ще не є критичною. Але коли мова заходить про те, щоб візуалізувати граф навіть із декількома десятками вершин та ребер, проблема розміщення стає першорядною. Реальні ж задачі, деколи потребують візуалізації графу, що містить десятки тисяч вершин та ще більше ребер [1, 2, 6].

Тому першим мірилом якості програмного засобу є реалізація якомога більше методів візуалізації графу. Звісно, реалізація більшості методів потребує дуже багато часу, тому в це ж мірило слід віднести можливість розширення набору реалізованих методів.

З першого мірила природно походить друге – програмний засіб має бути спроектовано таким чином, щоб його можливо було розширювати. Тобто щоб реалізація, наприклад, нових методів візуалізації не потребувала переробки загальної моделі [4].

Дуже часто, інформація, що має бути представлена графічно у вигляді графу, є результатом роботи іншої інформаційної системи. Це можуть бути дані з якоїсь бази даних, наприклад, інформація про ієрархію персоналу підприємства, або результат роботи програмних систем з маркетингових досліджень, тощо. В будь-якому разі, коли мова йде про взаємодію різних інформаційних систем, дуже гостро постає питання як саме має бути спроектована система, щоб її можливо було використовувати інтегровано з іншою [2]. Отже наступне мірило якості – можливість та якість організації взаємодії між системою, що розробляється, та іншими системами, які потенційно можуть бути використані для рішення різних етапів однієї задачі.

Як відомо, кожна складна система потребує підтримки. Система, що розробляється також не я виключенням, тому наступним мірилом слід вважати те, наскільки просто систему підтримувати [4].

Наступними мірлами слід виділити підтримку програмним засобом роботи із різними кодуваннями, в тому числі із кирилицею, та можливість роботи системи на різних операційних системах.

Останнім мірилом слід виділити так звану дружність до користувача, що мовою оригіналу називаються user-friendly. Під цим терміном мається на увазі зручність користування системою.

Отже кінцевий список мірил якості програмного засобу має виглядати так:

- можливість розширення списку реалізованих методів;
- розширюваність програмного засобу;
- можливість та якість організації взаємодії за іншими системами;
- зручність підтримки;
- робота із різними кодуваннями, в тому числі із кодуваннями кирилиці;
- зручність користування засобом.

**Вибір шаблону проєктування.** Для досягнення найвищих показників якості згідно із такими визначеними мірлами, як розширюваність та зручність підтримки доцільно звернутись до шаблону проєктування Model-View-Controller (MVC)

Основна ідея цього шаблону полягає в тому, щоб відокремити бізнес-логіку від інтерфейсу користувача та в результаті отримати систему, в якій можливо змінювати представлення, не змінюючи поведінку, та навпаки [4].

Згідно термінології MVC:

- модель (model) складає методи доступу до даних Системи, що зберігаються у зовнішньому джерелі або у самій моделі;
- представлення (view) відповідає за інтерфейс користувача (кнопки, поля для вводу тексту, таблиці, графіки, тощо),

- контролер управляє бізнес-логікою та взаємодією користувача, що діє через представлення, та даними.

Проста діаграма, що відображає взаємодію між моделлю, представленням та контролером представлена на рис 1. На діаграмі суцільними стрілками позначено безпосередній зв'язок, а пунктиром – опосередкований [4].

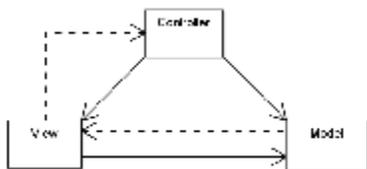


Рис. 1. Взаємодія між моделлю, представленням та контролером.

Найчастіший варіант потоку управління у MVC складаються з таких етапів [4]:

- а) користувач вступає у взаємодію з інтерфейсом користувача, наприклад, натискає на кнопку;
- б) контролер оброблює вхідну подію (event) від інтерфейсу користувача через обробник (handler);
- в) контролер повідомляє модель про дію користувача, що, можливо, призводить до зміни по моделі;
- г) представлення опосередковано використовує модель для зміни інтерфейсу користувача
- д) інтерфейс користувача очікує на подальші дії користувача, які, в свою чергу, починають цикл спочатку.

Шаблон MVC дозволяє мати декілька незалежних представлень моделі.

Побудування моделі. З ціллю підвищення ефективності розробки моделі програмного засобу візуалізації графів, було прийнято рішення використовувати спеціальну мову моделювання – UML (unified modeling language) [5].

Отже, для опису функціональної частини програмного засобу та варіантів його використання було розроблено діаграму варіантів використання (use-case diagram). В ході аналізу було визначено, яким функціоналом має володіти програма для забезпечення повноцінної роботи аналітика:

- а) відображення всього графу;
- б) відображення найкоротших шляхів між двома вибраними користувачем вершинами;
- в) відображення повного графу, побудованого відносно вибраної користувачем вершини та обмеженого вибраною користувачем глибиною;
- г) у користувача має бути можливість проглядати весь список вершин, які присутні у графі;
- д) редагування вершин та ребер;

- е) масштабування;
- ж) можливість збереження та завантаження графу у форматі XML;
- з) додання до графу нових вершин та зв'язків;
- і) видалення вершин та ребер;
- к) управління відображенням імен та деяких інших властивостей вершин і ребер на графіку;
- л) можливість зміни розташування вершин самим користувачем так, як він того забажає.

В результаті аналізу задачі виявлено три ключові сутності – вершина, ребро та графік, на якому граф має відобразитися. Також було прийнято рішення, що сам граф не варто розглядати як сутність тому, що усі властивості графу повністю відображаються його складовими, тобто вершинами та ребрами.

Згідно із прийнятим шаблоном MVC та визначеними атрибутами сутностей, була розроблена діаграма класів [5]. У спрощеному вигляді діаграма представлена на рис 2.

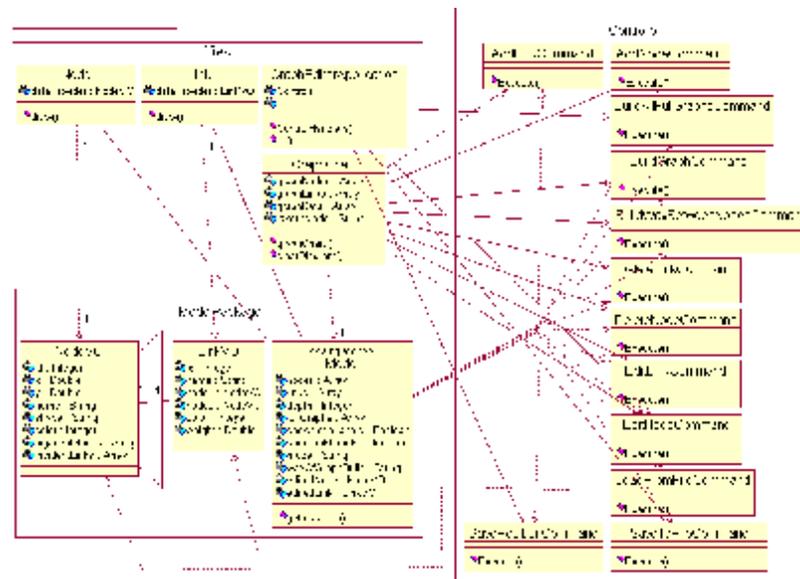


Рис. 2. Діаграма класів

**Висновки.** Розроблена модель програмного засобу та її програмна реалізація є крос-платформною, підтримує роботу з кириличними шрифтами, використовує стандартизований формат даних XML. Завдяки використанню шаблону проектування MVC розроблена модель також відповідає вимогам

розширюваності та зручності підтримки. Таким чином виконано усі вимоги, які були заявлені у меті.

**Список літератури:** 1. *Апанович З.В.* От рисования графов к визуализации информации. Новосибирск: препринт, 2007 2. *Апанович З.В.* Средства для работы с графами большого объема: построение и оптимизация компоновочных планов // Системная информатика: Сб.науч. тр. – Новосибирск: Изд-во СОРАН, 2006. – Вып. 10. Методы и модели современного программирования. – С. 7–58. 3. *Касьянов В.Н., Евстигнеев В.А.* Графы в программировании: обработка, визуализация и применение. – СПб.: БХВ-Петербург, 2003. – 1104 с. 4. Catalyst - Web Framework [<http://www.catalystframework.org/>] 5. *Г. Буч, Д. Рамбо, А. Джекобсон.* Язык UML. Руководство пользователя, Москва: Изд-во. ДМК, 2006. 6. *Balzer M., Deussen O., Lewerenz C.* Treemaps for the Visualization of Software metrics// ACM Symp. on Software Visualization (SoftVis), 2005. 7. *М.Ф. Бондаренко, Е.А. Соловьева, С.И. Маторин.* Основы системологии: Учебн. Пособие. Харьков: Изд-во Харьк. техн. Ун-та радиоэлектр., 1998, ISBN 5-7763-92217.

*Надійшло у редакцію 15.12.09*

УДК 005; 007; 681.5

*Е. А. СОЛОВЬЕВА*, д-р техн. наук, зав. Кафедрой СИ ХНУРЭ,  
*М. Ю. ФРОЛОВ*, аспирант ХНУРЭ

## **ОРГАНИЗАЦИЯ ЛОГИЧЕСКОЙ СТРУКТУРЫ ЭЛЕКТРОННЫХ АРХИВОВ НА ОСНОВЕ ЕСТЕСТВЕННОЙ КЛАССИФИКАЦИИ**

Проводиться аналіз можливості використання системологічного класифікаційного аналізу (СКА) на основі природної класифікації (ПК) у якості основи логічної структури організації електронних архівів для розв'язання деяких проблем організації дослідницького пошуку в електронних архівах.

Проводится анализ возможности использования системологического классификационного анализа (СКА) на основе естественной классификации (ЕК) в качестве основы логической структуры организации электронных архивов для решения некоторых задач организации исследовательского поиска в электронных архивах.

This article examines the possibility of using systemological classification analysis (SCA) on the basis of natural classification (NC) as the basis in the logical structure of the electronic files organization to solve some problems of research in search of electronic archives.

**Постановка проблемы.** Сегодня наблюдается небывалый подъем комплексной компьютеризации предприятий, корпораций и целых отраслей, причем первостепенная роль отводится построению автоматизированных систем документооборота и делопроизводства. Однако с течением времени оказалось, что внедрив мощные системы управления электронными документами, многие корпорации не получили ожидаемого эффекта. В чем причина? Огромное количество - до 80% - оперативной и справочной

информации по-прежнему остается на бумажных носителях и складывается в библиотеках или архивах [1]. Ручной поиск и обработка необходимых документов в таких бумажных хранилищах становится узким звеном функционирования многих крупных корпораций. Одним из возможных решений является внедрение и использование электронного документооборота, в том числе электронного архива (ЭА).

Электронные архивы повышают эффективность работы с корпоративными информационными ресурсами, в частности управленческой, финансовой, научно технической и проектно-конструкторской документацией. Эти документы зачастую существуют только в бумажном виде, плохо структурированы, хранятся в разных базах данных, на компьютерах отдельных пользователей и др.

Электронный архив – это единое структурированное хранилище электронных документов, на основе которого создается законченная система документооборота предприятия [2, 3, 4 и др.].

Электронный архив выполняет множество различных функций по работе и управлению документами, но для конечного пользователя принципиально важно исполнение поисковых функций.

В зависимости от объема хранимых документов различают три типа ЭА:

1. Малые архивы - характеризуются наличием небольшого числа хранимых документов (обычно не более 100 тысяч единиц). Общий объем архива до 500 Гб [5].
2. Средние архивы - количество документов от 100 000 до 2 000 000. Общий объем архива от 0,5 до 5 Тб [5].
3. Большие архивы - количество документов в архиве более 2 000 000. Общий объем архива более 5 Тб [5].

Существует две основных вида поиска документов в электронных архивах – атрибутивный и полнотекстовый поиск. Обычно малые архивы содержат атрибутивный поиск - каждому документу присваивается набор определенных атрибутов, присваиваемых документу во время его размещения в архиве. В дальнейшем документ ищется на совпадение значений этих атрибутов полям запроса [6, 7]. К атрибутам документа относят имя документа, время создания, автор, тип документа и т.п. Очевидно, что список таких атрибутов можно расширить. Совокупность атрибутов документа называется карточкой документа[7]. Эти поля могут заполняться произвольно или из predetermined справочников.

Средние и большие типы архивов помимо атрибутивного поиска содержат еще и полнотекстовый - обрабатывается все содержание документа, и затем по любому слову, входящему в данный документ, можно найти сам документ. Существует комбинация полнотекстового и атрибутивного поиска, когда атрибуты документа обрабатываются так же, как все содержание документа [7].

Однако, несмотря на сегодняшнее развитие информационных технологий средств поиска в ЭА, существуют некоторые проблемы, которые и будут рассмотрены и проанализированы в данной статье.

**Анализ последних исследований и публикаций.** Анализ организации поиска в ЭА, проводимый в данной работе, опирается на материалы основных фирм-производителей ЭА АйТи, IBM, IC, Optima, Евфрат и др., содержащие общую информацию о продукте, основные функции, круг решаемых задач, технические характеристики и т.п.

**Формулировка цели статьи.** Целью данной статьи является проведение анализа возможности использования метода системологического классификационного анализа (СКА) на основе естественной классификации (ЕК)[8] для построения логической структуры электронных архивов для решения некоторых проблем организации поиска в ЭА.

**Изложение основного материала.** Существует два подхода к поиску документов. Первый подход состоит в том, что в процессе поиска ищется документ, который точно существует в системе, и задача состоит в том, чтобы процесс поиска свелся к нахождению требуемого документа или документов – точный поиск. Второй подход состоит в том, что ищутся все документы по интересующему вопросу – исследовательский поиск [7]. Для последнего подхода присуще такое понятие, как шум – часть документов в результирующем наборе, где встретились подобные по синтаксису, но не по смыслу слова [7], т.е. документов нерелевантных поисковому запросу. Существует два основных типа поиска. Атрибутивный - каждому документу присваивается набор определенных атрибутов, присваиваемых документу во время его размещения в архиве [7]. В дальнейшем документ ищется на совпадение значений этих атрибутов полям запроса. Второй тип поиска, носит название полнотекстовый - автоматически обрабатывается все содержание документа, и затем по любому слову, входящему в данный документ, можно найти сам документ [7].

Соответственно существует зависимость между типами поиска и подходами к поиску. Для поиска известного документа, очевидно, применим атрибутивный поиск, для исследовательского поиска - полнотекстовый.

Рассмотрим ситуацию проведения исследовательского поиска для всех типов архивов. Ниже на рис. 1 показана схема организации поиска в малых ЭА.

Как было отмечено выше, для исследовательского поиска атрибутивный не подходит и, следовательно, остается навигация по логической структуре ЭА. Здесь и возникает одна из проблем поиска. Так как в качестве логической структуры выступает иерархическая структура, то при малом количестве уровней это делает понятной навигацию - пользователь представляет, где искать нужный документ. Однако в случае большого количества ветвей иерархии и их глубины эффективность и скорость поиска снижается, так как каждому новому уровню может соответствовать свое основание деления.

Одним из возможных способов устранения данных недостатков является использование естественной классификации в качестве логической структуры ЭА, что обеспечит единое основание деления для всей классификации [8].

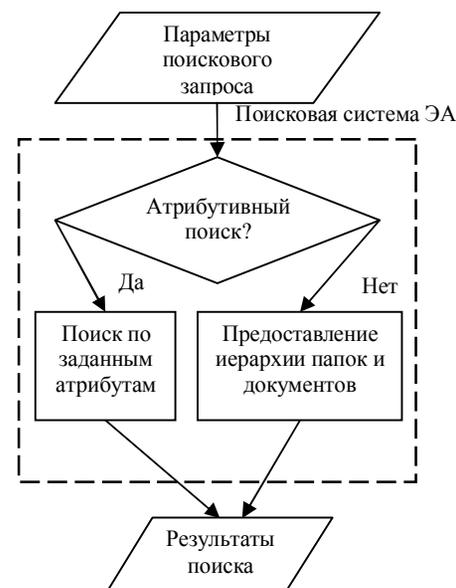


Рис. 1. Схема организации поиска в малых ЭА

В случае со средними и большими ЭА общая схема исследовательского поиска документов выглядит как показано на рис. 2.

Здесь появляется еще одна проблема, связанная с поиском в ЭА. Для достижения эффективного результата при использовании полнотекстового поиска необходимо составить корректный запрос к поисковой системе электронного архива. Но даже при условии отлично составленного запроса конечный результат поиска будет содержать так называемый «шум», т.е. документы нерелевантные запросу. Это происходит отчасти из-за того, что система поиска просматривает всю базу данных индексов ЭА. Используя метод системологического классификационного анализа для построения параметрической классификации в качестве логической структуры, мы получим для каждого документа свое главное отличительное функциональное свойство [8]. При исследовательском поиске функциональное свойство, по сути, является целью данного поиска. Пользователь, задав его в параметрах запроса, указывает необходимую ветвь иерархии для поиска. Данную ветвь можно использовать для дополнительного отсеивания документов, полученных в результирующем

наборе, оставляя только те документы, которые находятся в указанной ветви иерархии, что также сможет снизить уровень шума в конечном результате.

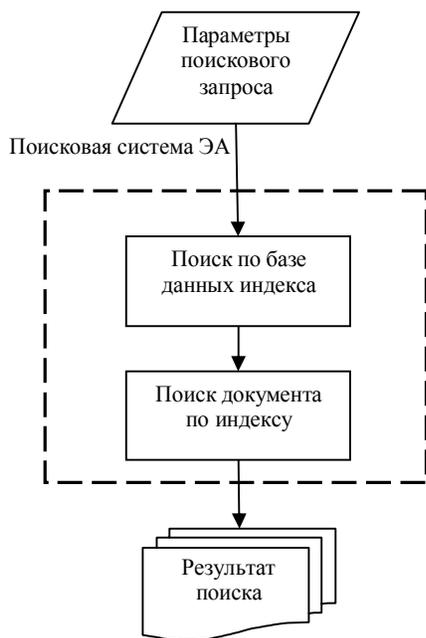


Рис. 2. Схема исследовательского поиска в средних и больших ЭА

**Выводы.** При использовании естественной классификации в качестве логической структуры ЭА появляется возможность частично решить некоторые проблемы, связанные с исследовательским поиском в электронных архивах: снижение шума в конечном результирующем наборе файлов в больших и средних архивах, а также повышение эффективности исследовательского поиска в малых архивах при развитой структуре цепочек документов.

**Список литературы:** 1. «Концепция построения электронных архивов», Алексей Марков [Электронный ресурс]. – Режим доступа Интернет: [http://www.alectarchive.ru/page.jsp?pk=node\\_1129899705713](http://www.alectarchive.ru/page.jsp?pk=node_1129899705713). 2. Электронный архив [Электронный ресурс]. – Режим доступа Интернет: <http://www.scanit.ru/archive/elar/>. 3. Концепция электронного архива [Электронный ресурс]. – Режим доступа Интернет: <http://www.mosarchiv.com/sub/service/useful/article/13>. 4. Обзор и анализ основных систем

автоматизации документооборота, Фролов М.Ю. Системы обработки информации, Харьков, 2009 – 131-134 стр. 5. Электронные архивы [Электронный ресурс]. – Режим доступа Интернет: <http://www.korusecm.ru/solutions/elib/>. 6. ProjectWise как система управления проектными данными, электронный архив, документооборот Алексей Лихачев «САПР и графика» 3'2001 [Электронный ресурс]–Режим доступа Интернет:<http://www.sapr.ru/article.aspx?id=6971&iid=285>. 7. Концепция построения систем автоматизации документооборота, Анатолий Гавердовский, АО ВЕСТЬ, Открытые системы. #1/97, стр. 29-34. 8. Соловьева Е.А. Естественная классификация: системологические основания - Харьков: ХНУРЕ, 1999 - 222 с.

Поступила в редколлегию 15.12.09

УДК 681.518

**М. І. БЕЗМЕНОВ**, канд. техн. наук, профессор НТУ «ХПІ»,  
**О. М. ЛАНСЬКИХ**, магістрант НТУ «ХПІ»,  
**В. Г. БОРИСОВ**, канд. техн. наук, доцент НТУ «ХПІ»

### МЕТРИКИ ЯК ОЦІНКА МОДЕЛЕЙ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ МЕДИЧНОГО ОБЛАДНАННЯ

Розглянуто види моделей якості програмного забезпечення. Обговорено важливість оцінки моделі якості програмного забезпечення у розробці медичних обладнань. Розглянуті метрики програмного забезпечення як кількісні показники оцінки моделі якості. Наведений приклад найбільш поширеної метрики з коду. Зроблено висновки.

Рассмотрены виды моделей качества программного обеспечения. Обсуждена важность оценки моделей качества программного обеспечения в разработке медицинских приборов. Рассмотрены метрики программного обеспечения как качественные показатели оценки модели качества. Рассмотрен пример наиболее распространенной метрики по коду. Сделаны выводы.

Software quality models have been considered. The importance of software quality models evaluation in medical devices development has been discussed. Software metrics have been considered as qualitative indicators of software quality model. Conclusions are made.

### Вступ

Технологія розробки медичних приладів змінювалась упродовж десятиріч. Найбільший крок був зроблений під час 1960–1970 років, коли ставили якість та ефективність на все більш значуще місце. Від тоді і до сих пір вчені працюють над методами покращення якості та ефективності.

Існує три класи медичних приладів:

Клас 1 – несе мінімальну потенційну шкоду хворій людині.

Клас 2 – несе шкоду хворій людині, але не впливає на її життя, загальні заходи контролю недостатні для забезпечення безпеки та ефективності.

Клас 3 – використання медичного прилада, у випадку неякісної його розробки, може вбити людину.

Успішний розвиток медичного обладнання найбільш за все залежить від вдосконаленості та якості вбудованого в нього програмного забезпечення (ПЗ). Якість медичного прилада найбільш за все залежить від рішень, прийнятих на ранніх ступенях розробки програмного забезпечення. Модель якості повинна постійно бути у циклічному процесі збору даних, аналізу, ідентифікації ризику, приймання та виконання коригувальних дій на протязі усіх етапів розробки програмного забезпечення: аналіз вимог, планування, кодування, тестування та випуск [1].

### 1. Використання моделі якості

Якість ПЗ повинна оцінюватися з використанням моделі якості. Модель якості [2] визначають після розроблення вимог до якості готового ПЗ та проміжного продукту, – вона є сукупністю атрибутів якості ПЗ, класифікованих в ієрархічну деревоподібну структуру характеристик і підхарактеристик. Верхній рівень цієї структури складається з характеристик якості, а нижній – з атрибутів якості ПЗ. Ця ієрархія не є строгою, тому що деякі атрибути можуть входити більш ніж в одну характеристику. Частіше за все ця ієрархічна структура використовується у вигляді UML-діаграм (англ.: Unified Modeling Language – Уніфікована Мова Моделювання).

Розглянемо поняття моделі якості [3] як модель внутрішньої [4] та зовнішньої [5]. Визначають шість основних характеристик якості ПЗ: функціональність, надійність, зручність використання, раціональність, супроводжуваність і переносимість (табл. 1).

Таблиця 1

Основні характеристики моделі якості ПЗ

Внутрішня та зовнішня якість					
Функціональність	Надійність	Зручність використання	Раціональність	Супроводжуваність	Переносимість
1) Функціональна повнота 2) Правильність 3) Здатність до взаємодії 4) Захищеність 5) Узгодженість функціональності	1) Безвідмовність 2) Стійкість до відхилень 3) Відновлюваність 4) Узгодженість надійності	1) Зрозумілість 2) Придатність до вивчення 3) Зручність інтерфейсу 4) Привабливість 5) Узгодженість зручності використання	1) Часова раціональність 2) Використовуваність ресурсів 3) Узгодженість раціональності	1) Аналізованість 2) Змінюваність 3) Стабільність 4) Тесто-придатність 5) Узгодженість супроводжуваності	1) Адаптованість 2) Налаштовуваність 3) Сумісність 4) Взаємозамінність 5) Узгодженість переносимості

Розглянемо докладніше кожну з характеристик моделі якості програмного забезпечення.

**Функціональність** – здатність ПЗ забезпечити функції, які виконують заявлені потреби та потреби, що мають на увазі при використанні ПЗ в заданих умовах. Вона включає такі функції:

- функціональна повнота – здатність ПЗ забезпечити відповідний набір функцій для заданих завдань та цілей користувача;

- правильність – здатність ПЗ забезпечувати вірні або припустимі результати чи дії з необхідним ступенем точності;
- здатність до взаємодії – здатність ПЗ взаємодіяти із зазначеними системами;
- захищеність – здатність ПЗ забезпечити захист інформації та даних (при їхньому зберіганні і передачі) таким чином, що неавторизовані користувачі та системи не будуть мати можливості до їх читання і модифікації, у той час як авторизованим користувачам і системам не буде відмовлено у доступі до них;
- узгодженість функціональності – здатність ПЗ до дотримання відповідних стандартів, угод, положень законів або подібних рекомендацій, що стосуються функціональності.

**Надійність** – група властивостей, що обумовлює здатність програмного забезпечення зберігати працездатність та перетворювати вихідні дані в очікуваний результат у заданих умовах за встановлений час. Ця властивість включає такі підхарактеристики:

- безвідмовність – здатність ПЗ уникати відмов (функціонування без відмов), які є результатом наявності дефектів у ПЗ;
- стійкість до відхилень – здатність ПЗ підтримувати необхідний рівень працездатності у випадках прояву програмних дефектів або порушення його інтерфейсу;
- відновлюваність – здатність ПЗ відновлювати заданий рівень працездатності, а також відновлювати дані, які безпосередньо ушкоджені у випадку виникнення відмови, після перезапуску ПЗ (автоматичного або оператором);
- узгодженість надійності – здатність ПЗ дотримуватись відповідних стандартів, угод, положень законів або подібних рекомендацій, що стосуються надійності.

**Зручність використання** – здатність ПЗ бути зрозумілим, придатним до вивчення і привабливим для користувача при його використанні в заданих умовах. Вона включає такі підхарактеристики:

- зрозумілість – здатність ПЗ забезпечити користувачеві зрозумілість того, чи може ПЗ бути використано і яким саме чином, для конкретних завдань і умов застосування;
- придатність до вивчення – здатність ПЗ до надання користувачеві можливості його вивчення;
- зручність інтерфейсу для користування – здатність ПЗ до надання користувачеві можливості управління та контролю за його роботою;
- привабливість – здатність ПЗ бути привабливим для користувача (відноситься до графічного інтерфейсу);

- узгодженість використання – здатність ПЗ до дотримання відповідних стандартів, угод, положень законів або подібних рекомендацій, що стосуються використання.

*Раціональність* – здатність ПЗ забезпечувати відповідну (допустиму) продуктивність із урахуванням займаних ресурсів у заданих умовах. Її підхарактеристиками є:

- часова раціональність – здатність ПЗ забезпечити відповідний (допустимий) час відгуку, обробку та пропускну здатність при виконанні його функцій у заданих умовах;
- використовуваність ресурсів – здатність ПЗ використовувати відповідну (допустиму) кількість і тип ресурсів при виконанні його функцій у заданих умовах;
- узгодженість раціональності – здатність ПЗ до дотримання відповідних стандартів або подібних рекомендацій, що стосуються ефективності.

*Супроводжуваність* – здатність ПЗ до модифікації, що може містити у собі виправлення, поліпшення або адаптацію ПЗ до змін середовища, вимог або функціональних специфікацій. Вона включає такі підхарактеристики:

- аналізованість – здатність ПЗ до діагностування дефектів або причин відмов ПЗ, а також ідентифікації частин програми, які мають потребу в корекції;
- змінюваність – здатність ПЗ до надання можливості для виконання заданої модифікації його коду, структури та алгоритмів або програмної документації;
- стабільність – здатність ПЗ до запобігання неочікуваних ефектів від модифікацій;
- тестопридатність – здатність ПЗ до валідації його змін (модифікацій);
- узгодженість супроводжуваності – здатність ПЗ до дотримання відповідних стандартів або подібних рекомендацій, що стосуються супроводжуваності.

*Переносимість* – здатність ПЗ бути перенесеним з одного організаційного, апаратного або програмного середовища в інше, що включає такі підхарактеристики:

- адаптованість – здатність ПЗ бути адаптованим до різних середовищ без застосування дій або засобів, відмінних від тих, що передбачено для цієї мети в розглянутому ПЗ, адаптованість містить у собі масштабованість внутрішніх елементів ПЗ (екранних полів, таблиць, форматів звітів тощо);
- налагоджуваність – здатність ПЗ до інсталяції в заданому середовищі;
- сумісність (безконфліктність) – здатність ПЗ без конфліктів співіснувати з іншим незалежним програмним забезпеченням у загальному середовищі, що спільно використовує загальні ресурси;

- взаємозамінність – здатність ПЗ до використання замість іншого заданого ПЗ з тією ж метою та у тому ж середовищі;
- узгодженість переносимості – здатність ПЗ до дотримання відповідних стандартів або подібних рекомендацій, що стосуються переносимості.

Модель якості у використанні визначає чотири характеристики, наведені на рис. 1.

Розглянемо характеристики моделі якості у використанні більш детально.

*Ефективність* – це здатність програмного забезпечення дозволяти користувачеві досягати зазначених цілей з точністю та повнотою в заданому контексті використання.

*Продуктивність* – це здатність програмного забезпечення дозволяти користувачеві витратити відповідну кількість ресурсів щодо ефективності, досягнутої в заданому контексті використання.



Рис. 1. Модель якості у використанні

*Безпека* – це здатність програмного забезпечення досягати прийнятних рівнів ризику шкоди людям, бізнесу, ПЗ, майну або навколишньому середовищу в заданому контексті використання.

*Задоволеність* – це здатність програмного забезпечення задовольняти користувачів у заданому контексті використання.

## 2. Застосування характеристик якості програмного забезпечення

Для ефективного вибору та застосування характеристик (показників) якості ПЗ [7], що представлені у попередньому розділі, їх необхідно класифікувати за типами на:

- описові – вони описують набір засобів і загальні характеристики об'єкта, його функції, безпеку, захищеність і важливість;
- кількісні – їх можна виміряти та чисельно зіставити з вимогами;
- якісні – ті, що визначають експертним методом.

Кількісний метод, як оцінка якості ПЗ, є одним з найпоширеніших. Кількісні величини повинні вибиратися та фіксуватися і супроводжуватися методикою їх чисельних вимірів при випробуваннях. Для кожної з них повинен бути встановлений ряд допустимих чисельних значень та шкали характеристик надійності. Одним із найефективніших засобів оцінки якості кількісним методом є використання метрик.

### 3. Метрики якості програмного забезпечення

Метрика програмного забезпечення [8] – це міра, що дозволяє отримати чисельне значення деякої властивості програмного забезпечення або його специфікації. Процес вибору та встановлення метрик є найважливішим.

#### 3.1. Процеси вибору та встановлення метрик

На першому етапі проводиться вибір та обґрунтування вихідних даних, що характеризують загальні особливості і етапи життєвого циклу проекту ПЗ та його споживачів, кожний з яких впливає на певні характеристики якості комплексу програм. Далі проводиться вибір, встановлення та затвердження конкретних метрик і шкал вимірювання показників якості проекту для їх наступного оцінювання і зіставлення з вимогами у процесі кваліфікаційних випробувань або сертифікації на певних етапах життєвого циклу програмного забезпечення.

Для якісних характеристик визначають та зафіксують у специфікаціях опис умов, за яких необхідно вважати, що дана характеристика реалізується в ПЗ. Результати аналізу і вибору номенклатури та метрик характеристик якості проекту ПЗ повинні бути задокументовані в специфікаціях вимог, погоджені з їх споживачами та затверджені замовником проекту.

#### 3.2. Вибір і використання метрик якості програмного забезпечення

Засоби вибору та використання метрик містять у собі послідовні реалізації наступних процедур:

- 1) визначення цільових характеристик якості ПЗ;
- 2) визначення складу метрик, що оцінюють цільові характеристики якості ПЗ;
- 3) підготовка до оцінювання;
- 4) задавання цільових значень і шкал допусків:
  - a) визначення необхідної програмної документації та складу вхідних параметрів для розрахунку метрик;
  - b) уточнення фактичного складу метрик, що використовуються;
  - c) визначення того, на якому етапі життєвого циклу програмного забезпечення використання яких метрик буде найбільш корисним;
  - d) оцінювання вхідних параметрів і розрахунок метрик;
- 5) аналіз результатів оцінювання;
- 6) узагальнення результатів оцінювання, побудова і згортка метричних діаграм;
- 7) порівняння розрахованих значень із заданими цільовими значеннями;
- 8) аналіз фактичної повноти та імовірності оцінювання;
- 9) формування експертного висновку (існуючих ризиків) за результатами оцінювання;
- 10) створення плану дій з метою рішення проблем та видалення ризиків, а також оцінювання термінів отримання результатів після виконаних корегуючих дій.

Зазвичай, на етапі підготовки оцінювання, при визначенні шкал допусків, використовують три зони які далі використовуються на етапі аналізу:

- 1) жовта – говорить про наявність допустимих значень, тобто та частина програмного забезпечення, на яку були накладені метрики, має задовільні результати та може бути виключена з наступного кроку аналізу якості;
- 2) зелена – знаходиться на границі між зеленою та червоною зонами в залежності від складності проекту та може бути включена на наступному кроці аналізу;
- 3) червона – вказує або на наявність проблем і ризиків у тій частині програмного забезпечення, на яку були накладені метрики (у цьому випадку обов'язково розглядається на етапі аналізу), або на роботу з невірними даними.

Існує величезна кількість метрик для оцінювання якості програмного забезпечення, серед яких можна виділити такі:

- 1) метрики коду;
- 2) метрики дефектів;
- 3) метрики надійності;
- 4) метрики стійкості до відхилень;
- 5) метрики зрозумілості;
- 6) метрики придатності до вивчення;
- 7) метрики захищеності;
- 8) метрики переносимості та багато інших.

Задачею аналітика є вибір лише тих метрик, які найбільш за все підходять до конкретної специфіки проекту. Звісно, найбільша увага завжди приділяється якості написання коду програмного забезпечення.

Одна з найпоширеніших метрик з коду є цикломатична складність коду.

*Цикломатична складність* (1) частини вихідного коду [8] – це кількість лінійно незалежних шляхів всередині коду. Наприклад, якщо вихідний код не містить точок прийняття рішень (наприклад, оператор IF або цикл FOR), складність дорівнюватиме одиниці, тому що існує тільки один можливий шлях крізь програмний код. Якщо програмний код має один оператор IF, який містить тільки одну умову, програма буде мати два можливі шляхи крізь код, а саме, перший шлях, де оператор IF оцінюється як TRUE, та другий шлях, де оператор IF оцінюється як FALSE.

Цикломатична складність коду, як правило, розраховується шляхом створення графа ресурсного коду. Вершинам цього графа відповідають рядки коду, а ребрами є можливі шляхи коду.

Цикломатична складність розраховується так:

$$M = E - N + P + R, \quad (1)$$

де  $M$  – цикломатична складність;

$E$  – кількість вершин графа;

$N$  – кількість ребер графа;  
 $P$  – кількість з'єднаних компонентів;  
 $R$  – кількість зворотних стверджень.

Частіше за все ця метрика збирається автоматично за допомогою спеціальних програм, наприклад, таких як Source Monitor. Наступним кроком буде визначення шкал допусків для подальшого аналізу даних (табл. 2).

Таблиця 2

Шкали допусків

Жовта зона	Зелена зона	Червона зона
Менше 3	Більш 3 та менше 6	Більш 6

Розглянемо графік значень отриманої метрики, які збиралися протягом п'яти місяців упродовж постійного процесу розробки програмного забезпечення (рис. 2).

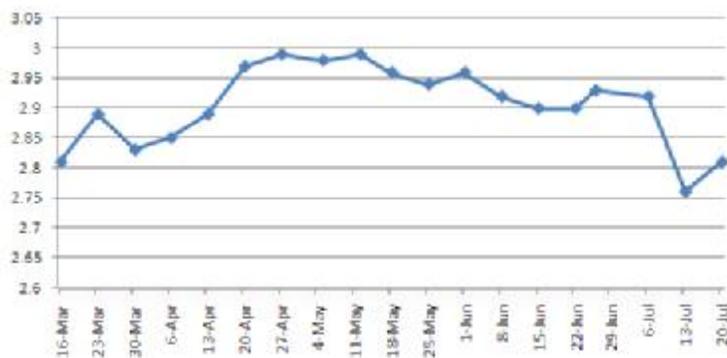


Рис. 2. Графік цикломатичної складності програмного коду

Загальна тенденція значення цикломатичної складності, як це показано на графіку, не переходить у зелену зону. Але у період з березня по травень відбувався видимий зріст цикломатичної складності, значення якої наближувалось до зеленої зони. Відповідні міри були прийняті відразу ж, як тільки ця тенденція була виявлена. Був проведений процес Code Review, у якому більш досвідчені програмісти проглядали та аналізували програмний код тих програмістів, які займалися розробкою ПЗ. Після цього було складено план з Code Refactoring (процес зміни програмного коду без зміни його загальної поведінки), де програмний код був перебудований та покращений. У результаті цих вжитих дій, починаючи з червня, спостерігається тенденція до спаду цикломатичної складності, як показано на графіку, що говорить про нормалізацію даних та про відсутність необхідності перебудовувати

програмний код. Тим самим, була попереджена якась частина дефектів ПЗ, адже ті класи чи функції, у яких значення цикломатичної складності перевищує пороги, є найбільш вірогідними джерелами дефектів.

### Висновки

Успішний розвиток медичного обладнання найбільш за все залежить від вдосконаленості та якості вбудованого в нього ПЗ. Оскільки кількісні оцінки виявилися настільки потужними в інших науках, у галузі інформатики було витрачено немало часу та зусиль на те, щоб розвинути аналогічні підходи у процесі розробки ПЗ. Том Де Марко, один із науковців, який перший описав метрики ПЗ як метод оцінки якості, якось заявив: «Ви не зможете контролювати те, що ви не можете виміряти» [1]. Ці слова підтверджують важливість використання метрик, як кількісний метод оцінки якості ПЗ.

Розглянутий приклад використання цикломатичної складності як метрики свідчить про необхідність аналізу процесу розробки програмного коду та його рефакторингу на протязі усіх етапів розробки програмного забезпечення, а саме під час аналізу вимог, планування, кодування, тестування та випуску. Приклад підтверджує можливість виявити аномалії та відхилення у програмному продукті ще у процесі кодування, а не вже на етапі тестування або навіть випуску. Це, в свою чергу, попереджає значний відсоток дефектів ПЗ у вбудованих медичних системах, адже дефекти ПЗ можуть призвести до неадекватної поведінки медичного прилада, що в свою чергу може негативно вплинути на життя людини.

**Список літератури:** 1. DeMarco Tom, Controlling Software Projects: Management, Measurement and Estimation 2002. – 279 p. 2. McCabe, A Complexity Measure 1976. – 154 p. 3. ISO/IEC 9126-1:2001 Software Engineering – Product Quality – Part 1: Quality model (Інжиніринг програмного забезпечення – Якість продукту – Частина 1: Модель якості). 4. ISO/IEC TR 9126-3:2003 Software Engineering – Product Quality – Part 3: Internal metrics (Інжиніринг програмного забезпечення – Якість продукту – Частина 3: Внутрішні метрики). 5. ISO/IEC TR 9126-2:2003 Software Engineering – Product Quality – Part 2: External metrics (Інжиніринг програмного забезпечення – Якість продукту – Частина 2: Зовнішні метрики). 6. ISO 9001:2008 Quality Management System – requirements (Системи менеджменту якості – вимоги). 7. Carnegie Mellon University, Mark C. Paulk, Bill Curtis. The Capability Maturity Model: Guideline for Improving the Software Process 2001. – 434 p. 8. ISO/IEC TR 9126-4:2004 Software Engineering – Product Quality – Part 4: Quality in use metric (Інжиніринг програмного забезпечення – Якість продукту – Частина 4: Метрики якості у використанні).

Надійшло у редколегію 29.11.09

А. А. БОЕВА, канд. физ.-мат. наук, ст. преподаватель НТУ “ХПИ”

### МОДЕЛИРОВАНИЕ ПОЛЯРИЗАЦИОННОГО ЭЛЕКТРИЧЕСКОГО ЗАРЯДА В ПЛАЗМЕ СМЕРЧА

У статті розглядається задача математичного моделювання електричного поля в плазмі смерчу - атмосферного вихору, що створюється сильним хмаровим електричним зарядом. Знайдено просторовий розподіл поляризаційного заряду в плазмі в випадку коли діелектрична проникність наближається до одиниці. Доведено, що цей заряд за величиною можливо порівняти зі слабким хмаровим крапельним зарядом.

В статье обсуждается задача математического моделирования электрического поля в плазме смерча - атмосферного вихря, создаваемого сильным облачным электрическим зарядом. Найдено пространственное распределение поляризационного заряда в плазме для случая, когда диэлектрическая проницаемость ее близка к единице. Показано, что этот заряд по величине сравним со слабыми облачными капельными зарядами.

In the paper the problem of mathematical modeling of electric field in thumping plasma i.e. atmospheric rotor created by strong cloud electric charge is discussed. The space distribution of polarized charge in plasma for the case when dielectric permeability is close to unity was found. It was shown that this charge is comparable in absolute value to weak cloud drop charges.

**Введение.** Газовые разряды в постоянных и переменных электрических полях находят широкое применение в науке и технике (спектроскопия, лазеры, сварка, источники света и др.) [1]. Важное место среди них занимают и атмосферные разряды, например, молнии различных типов, которые давно изучаются как теоретически, так и экспериментально [2]. Последнее время с газовыми разрядами, создаваемыми в атмосфере сильными облачными капельными зарядами, связывают и происхождение таких грозных природных явлений, как смерчи [3-6].

Характерной особенностью разрядов средних и высоких давлений является локальная зависимость параметров плазмы (концентрации заряженных частиц, их температуры) от напряженности электрического поля. В силу этого становятся локальными функциями напряженности электрического поля диэлектрическая проницаемость и проводимость плазмы, и задача определения поля в плазме разряда становится нелинейной. Подобные электродинамические задачи достаточно широко изучены в случаях, когда уравнения для поля однородные и одномерные (например, [7]). Постановка электродинамической задачи для смерча исключает такие варианты, т.к. плазма в нем создается вблизи облачного заряда и учитывает его пространственную структуру. В связи с этим, изучение пространственной и неоднородной нелинейной электродинамической задачи представляет большой интерес.

**Задача об электрическом поле в плазме смерча.** Согласно электромагнитной теории смерч представляет собой термодинамически неравновесный (далее, неравновесный) атмосферный газовый разряд, создаваемый сильным капельным зарядом, сосредоточенным обычно на краю грозового облака [8]. Пространственное распределение электрического потенциала  $\Phi$  и электрического поля  $\vec{E} = -\nabla\Phi$  в плазме разряда описывается следующим уравнением:

$$e\Delta\Phi + m(e-1)(\nabla \ln V \cdot \nabla\Phi) = -4pr_0(\vec{r}), \quad (1)$$

$$V = \sqrt{(\nabla\Phi)^2} / p, \quad m = Vd / dV(E_i / kT_E(V)). \quad (2)$$

Здесь  $p$  - давление газа,  $r_0(\vec{r})$  - объемная плотность облачного заряда, считающаяся заданной функцией радиус - вектора  $\vec{r}$ .

Величина  $V$  в (2) есть хорошо известный в теории газового разряда параметр  $E/r$ , характеризующий распределение энергии поля по степеням свободы атомов и молекул [1]. Для смерча из-за наличия в реальной атмосфере прилипания электронов к молекулам воды и кислорода  $V \geq 7.34$  В/см-Торр, вследствие чего атмосферно-облачный разряд является неравновесным и холодным [3-6]. Выражение для  $m$  в (2) определяется отношением потенциала ионизации газа  $E_i$  к тепловой энергии электронов  $kT_E$  ( $k$  - постоянная Больцмана,  $T_E$  - электронная температура), являющейся функцией  $V$ . Величина  $e$  в уравнении (1) есть диэлектрическая проницаемость плазмы. Она также зависит от параметра  $V$ .

Решение уравнения (1) должно стремиться к нулю при удалении от облачного заряда. На границе с землей при  $z=0$  оно должно удовлетворять крайевым условиям, соответствующим электрическим свойствам грунта. Например, для бесконечно проводящей поверхности тангенциальные компоненты электрического поля должны обращаться в нуль. Здесь возможны и другие варианты крайевых условий, учитывающих также и рельеф местности.

Диэлектрическая проницаемость  $e$  в зависимости от величины  $V$  может изменяться от достаточно больших по модулю отрицательных значений до  $\infty$  (атмосферном давлении и близким к нему  $e \approx 1$ . Причина этого кроется в большой частоте столкновений электронов с нейтральными частицами. В этом случае в уравнении (1) имеется малый параметр  $a = e - 1$  и задача может быть решена методом возмущений по этому параметру. Эта ситуация наиболее интересна в прикладном отношении. С увеличением высоты частота столкновений экспоненциально уменьшается, и здесь диэлектрическая проницаемость может стать отрицательной. В этом случае знак облачного

заряда как бы изменится на противоположный, т.к. роль заряда уже будет играть величина  $-r_0/|e|$ . В этой ситуации возможен и плазменный резонанс, когда в некоторой области неоднородной плазмы  $e(r_0^1) = 0$ . Для решения задачи здесь уже должен использоваться метод сингулярных возмущений [9].

**Плотность поляризационного заряда.** При значении диэлектрической проницаемости близком к единице решение рассматриваемой задачи строится в следующем виде:

$$\Phi = \Phi_0 + a\Phi_1 + a^2\Phi_2 + \dots, \quad a = 1 - e.$$

Потенциал нулевого приближения удовлетворяет при этом уравнению Пуассона

$$\Delta\Phi_0 = -4\pi r_0(r).$$

с соответствующими краевыми условиями.

Для потенциала первого приближения получим:

$$\Delta\Phi_1 = -4\pi r_1(r),$$

где  $r_1(r) = (1/4\pi)\text{div}(\nabla \ln V \cdot \nabla \Phi)_{\Phi=\Phi_0}$ .

Видно, что  $r_1(r)$  играет роль электрического заряда, индуцированного потенциалом  $\Phi_0$ , и нахождение потенциала  $\Phi_1$  требует знания пространственного распределения этой величины. По физическому смыслу  $r_1(r)$  есть заряд поляризации, возникающий в плазме под действием внешнего электрического поля.

Ниже проведено изучение пространственной структуры заряда  $r_1(r)$  для случая, когда облачный заряд моделируется бесконечно тонким круглым диском радиуса  $a$ , расположенным над землей на высоте  $h$ . Величина заряда равна  $Q$ .

Нахождение электрического потенциала нулевого приближения может быть проведено с помощью рядов по полиномам Лежандра [10]. Для сухой земли это решение в ближней зоне заряда имеет вид ( $R$  и  $q$  - сферические координаты, связанные с землей и вертикальной осью облачного заряда):

При  $R/a \leq 1$

$$\Phi_0 = \frac{2Q}{a} \left[ 1 - \frac{R}{a} P_1(\cos q) + \frac{1}{2} \left( \frac{R}{a} \right)^2 P_2(\cos q) - \frac{1}{8} \left( \frac{R}{a} \right)^4 P_4(\cos q) + \right.$$

$$\left. + \frac{1}{16} \left( \frac{R}{a} \right)^6 P_6(\cos q) - \dots \right].$$

Используя зависимость электронной температуры от напряженности электрического поля и потенциала ионизации из работы [3], представим плотность поляризационного заряда в следующем виде:

$$r_1 = \frac{1}{4\pi a} \frac{3}{4} \left( \frac{E_1}{kT_0} \right) E^{-2} \left( \frac{E_*}{E} \right)^{3/4} \left[ E_r^2 \frac{\partial E_r}{\partial \bar{r}} + 2E_r E_z \frac{\partial E_r}{\partial \bar{z}} + E_z^2 \frac{\partial E_z}{\partial \bar{z}} \right].$$

Здесь  $\bar{r} = r/a$ ,  $\bar{z} = z/a$  - координаты, связанные с землей,  $E_* = 0.14$  В/см - характерное электрическое поле, определяемое процессами неупругих столкновений электронов,  $E_r$  и  $E_z$ , соответственно, - радиальная и осевая компоненты электрического поля.

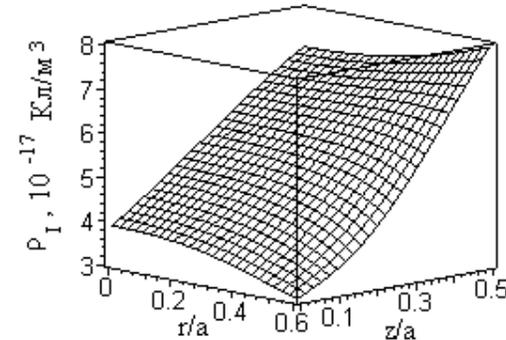


Рис.1 Плотность поляризационного заряда.  $Q = -64$  Кл,  $a = 1$  км,  $h = 0.5$  км

На рисунке представлена пространственная зависимость  $r_1$  под облачным зарядом  $Q = -64$  Кл, имеющим радиус  $a = 1$  км и расположенным на высоте  $h = 0.5$  км. Вычисления проведены только в области ближнего поля и там, где существует разряд [3]. Видно, что поляризационный заряд, как и положено, положительный, что указывает на эффект экранирования плазмой отрицательного облачного заряда. На оси плазменного столба ( $r = 0, 0 \leq z \leq h$ )  $r_1$  монотонно увеличивается с высотой, так как здесь его поведение определяется только положительной вертикальной компонентой электрического поля. При удалении от оси начинает играть роль уже и отрицательная радиальная компонента, ее величина и производные. По

величине плотность поляризованного заряда сравнима со слабыми капельными зарядами в облаках. [11]. Вместе с тем, она на пять - семь порядков меньше сильных облачных флуктуационных зарядов. Это факт также вполне объясним, поскольку при значении диэлектрической проницаемости, близком к единице, плазма поляризуется слабо.

**Выводы.** В работе обсуждается задача математического моделирования электрического поля в плазме смерча - атмосферного вихря, создаваемого сильным облачным электрическим зарядом. Найдено пространственное распределение поляризованного заряда в плазме для случая, когда диэлектрическая проницаемость ее близка к единице. Показано, что этот заряд по величине сравним со слабыми облачными капельными зарядами.

**Список литературы:** 1. Райзер Ю. П. Физика газового разряда.- М.: Наука, 1987.- 592 с. 2. Юман М. Молния.- М.: Мир,1972.- 328 с. 3. Боев А. Г. Плазменная теория смерча. // Вопросы атомной науки и техники. Плазменная электроника и новые методы ускорения.-2008.- № 4.- С. 133-138. 4. Боев А. Г. Газоразрядная теория смерча. // Доповіді НАНУ.- .2009.- № 4.- с.70- 75. 5. Боев А. Г. Электромагнитная теория смерча. I. Электродинамика вихря. // Радиофизика и радиоастрономия.- 2009.- Т.14, № 2.- С. 121-149. 6. Боев А.Г. Электромагнитная теория смерча.. II. Гидродинамика вихря. // Радиофизика и радиоастрономия.- 2009.- Т.14, № 3.- С. 233-253. 7. Басс Ф. Г., Гуревич Ю. Г. Горячие электроны и сильные электромагнитные волны в плазме полупроводников и газового разряда.- М.: Наука, 1975.- 400 с. 8. Наливкин Д. В. Смерчи.- М.: Наука, 1984.- 112.с. 9. Васильева А. Б., Бутузов В. Ф. Асимптотические методы в теории сингулярных возмущений.- М.: Высшая школа, 1980.- 208 с. 10. Морс Ф. М., Фейсбах Т. Методы теоретической физики. Т.2.- М.:ИЛ,1960.- 886 с. 11. Имянитов И. М., Чубарова Е. В. и др. Электричество облаков.- Л.: Гидрометеоиздат, 1971.- 92 с.

Поступила в редакцию 21.12.09

## СОДЕРЖАНИЕ

<b>Балота С. М., Безменова О. Н.</b> Квазистатическая математическая модель задачи управления технологическими параметрами прокатного стана .....	3
<b>Шатохина Н. В.</b> Моделирование эффективных параметров модели деятельности производственно-экономической системы .....	12
<b>Годлевский М. Д., Станкевич А. А.</b> Классификация иерархических систем управления и координации бизнес-процессов цепочек поставок .....	18
<b>Годлевский М. Д., Воловщиков В. Ю., Шано В. Ф.</b> Принципы построения траектории развития корпоративной информационно-вычислительной системы для случая ненаследуемых структур .....	24
<b>Shekhovtsov V. A.</b> Towards assessing simulated service qualities by business stakeholders outside the system boundary .....	29
<b>Баженов Н. А.</b> Модели управления качеством сбора требований к программному обеспечению на основе текстов спецификаций .....	35
<b>Соколов Б. Н.</b> Модели управления качеством обработки требований к программному обеспечению .....	43
<b>Маляр М. М.</b> Визначення та цілеспрямована зміна вагових коефіцієнтів при багатокритеріальному виборі .....	51
<b>Лобанова Л. С., Пасічник В. О., Черняк О. О.</b> Оптимізація обчислень в інтегральному методі найменших квадратів наближення функцій однієї та двох змінних .....	57
<b>Сейед Моджтаба Джафари Хенджани, Северин В. П.</b> Перспективы применения генетических алгоритмов для оптимизации следящего привода и систем управления паровой турбиной АЭС .....	63
<b>Кривенко Н. В.</b> Решение задачи выбора сценария развития для атомной электростанции.....	73
<b>Нескородова Т. В.</b> Логико-формальные правила формирования товарного обеспечения торгового предприятия .....	81
<b>Куц С. С., Куценко А. С., Марченко И. И.</b> Компьютерное моделирование формирования поверхности тонких плёнок кинетическим методом Монте-Карло .....	93
<b>Хацько Н. Е.</b> Оптимизация структуры минимально-избыточной системы неравноточных измерителей .....	100
<b>Усенский В. Б., Асютин А. Д.</b> Разработка имитационной модели движения инерциального блока на двухосном поворотном столе .....	107
<b>Ганжа А. М., Марченко Н. А.</b> Комп'ютерне моделювання процесів у складних теплообмінних апаратах.....	113
<b>Коваленко С. В., Коваленко С. Н., Бирюкова Н. В.</b> Анализ методов поиска сходных изображений среди больших коллекций графических файлов.....	121
<b>Безменов Н. И., Коваленко С. В., Ткачев М. А.</b> Использование нейронных сетей для распознавания графических образов .....	126

<i>Горбач Т. В., Святкин Я. В., Шубин И. Ю.</i> Методы интеграции учебных материалов для дистанционного обучения.....	131
<i>Любчик М. Л.</i> Аналіз показників якості функціонування системи «бонус-малус» на основі марківських моделей.....	140
<i>Стратієнко Н. К., Суццова К. С.</i> Підхід до розробки інформаційного забезпечення для вирішення задачі підбору персоналу в рекрутингових компаніях.....	144
<i>Кондращенко В. В., Москаленко В. В., Захарова Т. В.</i> Архитектура СППР для построения схемы финансирования инвестиционного проекта .....	149
<i>Кондращенко В. В.</i> Программная реализация СППР для построения схемы финансирования инвестиционного проекта .....	157
<i>Крючковский В. В., Брынза Н. А., Баддур А. Х.</i> Анализ адекватности взаимной трансформации неопределенностей при вычислении скалярных интервальных значений полезности альтернатив.....	169
<i>Соловйова К. О., Мовчан В. В.</i> Розробка моделі програмного засобу візуалізації мережевих структур.....	178
<i>Соловьева Е. А., Фролов М. Ю.</i> Организация логической структуры электронных архивов на основе естественной классификации.....	183
<i>Безменов М. Г., Ланських О. М., Борисов В. Г.</i> Метрики як оцінка моделей якості програмного забезпечення медичного обладнання .....	188
<i>Боева А. А.</i> Моделирование поляризованного электрического заряда в плазме смерча.....	197

## НАУКОВЕ ВИДАННЯ

### ВІСНИК НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ «ХП»

#### Збірник наукових праць

#### Тематичний випуск «Системний аналіз, управління та інформаційні технології»

*Випуск 9*

Наукові редактори М. Д. Годлевський, О.С.Куценко  
Технічний редактор М. Г. Безменов

Відповідальний за випуск І. Б. Обухова

Обл.-вид. № 72-10.

Підп. до друку 15.01.10 р. Формат 60×84 1/16. Папір офісний.  
RISO-друк. Гарнітура Таймс. Умов. друк. арк. 8,5. Облік.-вид. арк. 10,1.  
Наклад 300 прим. Перший завод 1-100. Зам. № 11. Ціна договірна.

Видавничий центр НТУ «ХП».  
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.  
61002, Харків, вул. Фрунзе, 21

Видруквано в друкарні ФОП Стеценко І. І.  
Свідоцтво про видавничу діяльність ХК № 82 від 22.04.2003 р.  
61019, Харків, пр. Ілліча, 103а, кв. 21,  
тел. (057) 758-17-35